



# Nouvelle génération de systèmes de vision temps réel à grande dynamique

Pierre-Jean Lapray

## ► To cite this version:

Pierre-Jean Lapray. Nouvelle génération de systèmes de vision temps réel à grande dynamique. Autre [cs.OH]. Université de Bourgogne, 2013. Français. NNT : 2013DIJOS032 . tel-00947744

**HAL Id: tel-00947744**

**<https://theses.hal.science/tel-00947744>**

Submitted on 17 Feb 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The background of the top section features a large, faint, light-gray logo that reads 'SPIM'. The 'S' is a stylized wave, the 'P' is a vertical bar, the 'I' is a vertical bar, and the 'M' is a large, stylized 'M' shape.

# SPIM

## Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques  
U N I V E R S I T É D E B O U R G O G N E

# Nouvelle génération de systèmes de vision temps réel à grande dynamique

 Pierre-Jean LAPRAY





# SPIM

## Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques  
UNIVERSITÉ DE BOURGOGNE

N° X X X

THÈSE présentée par  
**Pierre-Jean LAPRAY**

pour obtenir le  
Grade de Docteur de  
l'Université de Bourgogne

Spécialité : **Instrumentation et Informatique de l'Image**

## Nouvelle génération de systèmes de vision temps réel à grande dynamique

Soutenue le 18 octobre 2013 15h à Dijon devant le Jury :

Dominique GINHAC	Directeur de thèse	Professeur à l'Université de Bourgogne
Barthélémy HEYRMAN	Coencadrant	Maître de conférences à l'Université de Bourgogne
François BERRY	Rapporteur	Maître de conférences à l'Université de Clermont-Ferrand
Patrick GARDA	Rapporteur	Professeur à l'Université de Pierre et Marie Curie
Patrick MARQUIÉ	Examineur	Professeur à l'Université de Bourgogne
Raphaël DAVID	Examineur	Ingénieur de recherche, CEA LIST



## REMERCIEMENTS

Je tiens tout d'abord à remercier mon directeur de thèse Dominique Gin hac, professeur et directeur adjoint du laboratoire LE2I (Laboratoire Électronique, Informatique et Image) de l'université de Bourgogne. Je remercie aussi mon coencadrant Barthélémy Heyrman, maître de conférence au département électronique, qui m'a accompagné tout on long de ma thèse. Merci à vous de m'avoir fait confiance dans mes démarches, d'avoir considéré et soutenu mon travail, de m'avoir laissé l'entière liberté dans mes recherches, tout en étant disponibles à tout instant pour mes questionnements durant les trois dernières années. Je remercie Matthieu Rossé, pour ses qualités humaines, son travail effectué au sein du projet, et sans qui la construction des prototypes n'aurait pas pu se faire.

Je suis également reconnaissant à François Berry et Patrick Garda pour avoir bien accepté d'être les rapporteurs de la thèse. Je remercie Patrick Marquié et Raphaël David d'avoir bien voulu être les examinateurs du Jury.

Je tiens bien évidemment à remercier tout aussi profondément Julien Dubois, pour ses qualités humaines, ses conseils avisés et ses encouragements au cours de mon projet. Je remercie la société Thales Angénieux pour la collaboration que nous avons pu avoir au début de la thèse.

Je remercie également la société Eyenetics (Gerald Bardini, Pierre Chatelier et Yoan Marin), Romuald Jolivot, Camille Simon, Imen Charfi, Jean-Baptiste Thomas, Gilberto Ochoa, Sergio Martinez qui n'ont pas été seulement des collègues, mais aussi de véritables amis, avec qui j'ai pu passer des moments forts.

Je tiens aussi à saluer toute l'équipe administrative du laboratoire, le directeur Fabrice Mériaudeau, les secrétaires (Dounia Radi, Mélanie Arnoult, Dorielle-Agnès Batis et Nadia Bader), Khalil Jradi, Franck Marzanni, les gens qui m'ont encadré et conseillé pour mon monitorat (Denis Pellion, Saverio Morfu, Patrick Marquié), ainsi que mes collègues de bureau successifs.

Je tiens ensuite à saluer toutes celles et tous ceux que j'ai pu croiser durant ces quelques années au LE2I.

Je remercie bien évidemment mes parents, ma famille, Julie et mes amis, qui m'ont soutenu tout au long de ma thèse.

Merci à vous !



# SOMMAIRE

Remerciements . . . . .	5
<b>1 Introduction Générale . . . . .</b>	<b>11</b>
1.1 Principe de l'imagerie à grande dynamique . . . . .	11
1.2 Contexte de la thèse . . . . .	13
1.3 But de la recherche et problématique . . . . .	14
1.4 Structure du manuscrit . . . . .	16
<b>2 Algorithmie pour l'imagerie HDR . . . . .</b>	<b>19</b>
2.1 Introduction . . . . .	19
2.2 Définitions . . . . .	21
2.2.1 Grandeurs radiométriques et photométriques . . . . .	21
2.2.1.1 La radiométrie . . . . .	22
2.2.1.2 La photométrie . . . . .	23
2.2.2 Notations . . . . .	24
2.2.3 La gamme dynamique . . . . .	25
2.2.4 Les limites du capteur d'image . . . . .	27
2.3 Création d'image HDR à partir d'expositions multiples . . . . .	31
2.3.1 Fusion d'expositions dans le domaine de radiance . . . . .	31
2.3.1.1 Récupération de la courbe de réponse . . . . .	32
2.3.1.2 Construction de la carte de radiance . . . . .	35
2.3.1.3 Reproduction de tons (Tone Mapping) . . . . .	37
2.3.1.4 L'évaluation des opérateurs de reproduction de tons . . . . .	39
2.3.2 Fusion d'expositions dans le domaine image . . . . .	41
2.4 Problèmes liés aux vidéos HDR . . . . .	45
2.4.1 Alignement des images . . . . .	45
2.4.2 Détection et élimination de "ghost" . . . . .	46

2.5	Conclusion . . . . .	47
<b>3</b>	<b>État de l'art : Vers un système matériel temps réel dédié HDR</b>	<b>49</b>
3.1	Introduction . . . . .	49
3.2	Techniques pour étendre la dynamique d'un capteur . . . . .	49
3.3	Les systèmes à fusion d'exposition . . . . .	53
3.4	Les systèmes de capture . . . . .	56
3.5	Les systèmes de création de contenu HDR . . . . .	58
3.6	Les systèmes à reproduction de tons . . . . .	59
3.7	Les systèmes hardware . . . . .	61
3.7.1	Les systèmes non temps réel . . . . .	61
3.7.2	Les systèmes temps réels . . . . .	63
3.8	Choix matériel pour un système de vision HDR en temps réel . . . . .	66
3.8.1	Le capteur . . . . .	66
3.8.2	Le FPGA et le GPU : des architectures parallèles . . . . .	68
3.8.3	Un dispositif expérimental pour la vision à grande dynamique . . . . .	70
3.9	Conclusion . . . . .	71
<b>4</b>	<b>HDR-ARtiSt : une caméra HDR</b>	<b>75</b>
4.1	Introduction . . . . .	75
4.2	Système global . . . . .	76
4.3	Auto Exposition Multiple (MAE) . . . . .	77
4.3.1	Algorithmie pour la MAE . . . . .	80
4.4	Management de la mémoire dédié HDR-MMU . . . . .	84
4.4.1	Définition d'une vidéo temps réel . . . . .	84
4.4.2	Principe . . . . .	85
4.5	Algorithmie . . . . .	88
4.5.1	Un outil Matlab : <i>HDR-Lab</i> . . . . .	89
4.5.1.1	Génération HDR . . . . .	91
4.5.1.2	Déploiement de TMOs . . . . .	91
4.5.1.3	Métriques de qualité d'image . . . . .	92
4.5.1.4	Outils de visualisation . . . . .	93

4.5.2	Choix algorithmique pour la génération HDR . . . . .	94
4.5.2.1	Format de donnée . . . . .	94
4.5.2.2	Debevec et al. . . . .	98
4.5.2.3	Détermination de la courbe de réponse . . . . .	100
4.5.2.4	Choix de la fonction de pondération . . . . .	101
4.5.2.5	Simplifications . . . . .	102
4.5.3	Choix algorithmique pour le tone mapping . . . . .	103
4.5.3.1	Duan et al. . . . .	106
4.5.3.2	Simplifications . . . . .	107
4.5.3.3	Reinhard et al. . . . .	108
4.6	Conclusion . . . . .	110
<b>5</b>	<b>Implémentation et résultats du système HDR-ARtiSt</b>	<b>111</b>
5.1	Introduction . . . . .	111
5.2	Environnement de simulation Hardware . . . . .	111
5.3	Estimation des temps d'exposition avec la MAE . . . . .	112
5.4	Simulation de la MMU . . . . .	113
5.5	Simulation du système HDR-2 . . . . .	114
5.5.1	Debevec et al. . . . .	114
5.5.2	Limites du système HDR-2 . . . . .	115
5.6	Implémentation complète de HDR-2 . . . . .	117
5.6.1	Résultats de synthèse matérielle sur Virtex 5 . . . . .	117
5.6.2	Latence observable . . . . .	119
5.7	Implémentation de HDR-3 . . . . .	119
5.8	Extension par la couleur . . . . .	121
5.9	Expérimentations et conclusions . . . . .	123
5.9.1	Implémentation sur Virtex 6 . . . . .	123
5.9.2	<i>HDR-Viewfinder</i> : Application Ethernet avec Qt . . . . .	124
5.10	Comparaison avec les architectures existantes . . . . .	125
5.11	Perspective : le problème de Ghost . . . . .	127
5.12	Conclusion . . . . .	128



<b>6 Conclusion</b>	<b>131</b>
6.1 Principales contributions . . . . .	131
6.2 Perspectives . . . . .	132
6.3 Expérience personnelle . . . . .	132
<b>A Description du Memory Management Unit (MMU) pour P=3</b>	<b>135</b>
<b>B HDR-Lab</b>	<b>139</b>
<b>C Estimation de la courbe de réponse</b>	<b>141</b>
<b>D "HDR-Viewfinder" : les sorties du système HDR-ARtiSt</b>	<b>145</b>
<b>E Étalon or</b>	<b>147</b>
<b>F Détails des cartes capteur</b>	<b>149</b>
F.1 Liste des composants . . . . .	149
F.1.1 Carte Sous-Fille . . . . .	149
F.1.1.1 L'optique . . . . .	149
F.1.1.2 L'électronique . . . . .	149
F.1.2 Carte Fille . . . . .	149
F.1.2.1 L'électronique . . . . .	149

# INTRODUCTION GÉNÉRALE

## 1.1/ PRINCIPE DE L'IMAGERIE À GRANDE DYNAMIQUE

Un appareil photographique ou une caméra standard est capable de capturer seulement une partie de l'information d'une scène dans le domaine visible. Par contre, cette même scène peut être perçue pleinement par le système visuel humain (voir l'analogie avec la peinture Figure 1.1). Cela se vérifie plus particulièrement pour des scènes réelles où la différence d'intensité lumineuse entre les zones sombres et les zones très fortement éclairées est grande. Lorsque l'on capture une telle scène, le capteur standard n'est pas capable de stocker la gamme entière de dynamique de luminance pour chaque pixel. Il en résulte une image où la différence des valeurs de pixel n'est plus discernable pour des luminances relativement faibles (pixels saturés à 0) et les luminosités relativement hautes (pixels saturés à 1023 pour un capteur de format de sortie 10 bits). La technique d'imagerie destinée à produire des images à grande gamme dynamique à partir d'images "Low Dynamic Range" <sup>1</sup>, apparaît comme l'une des solutions les plus adaptées et les moins coûteuses à ce jour, afin d'augmenter la plage dynamique des imageurs numériques. À titre d'exemple, Apple a récemment introduit un mode HDRI sur ses caméras d'iPhone 4, capable de prendre plusieurs photos avec des temps d'exposition différents et de les fusionner pour pouvoir afficher un plus grand spectre d'intensité lumineuse. L'imagerie HDR encode les images avec un nombre de bits plus élevé que le simple format 24 bits utilisé pour encoder une image couleur standard. Il existe plusieurs façons de procéder afin d'obtenir une image avec une grande dynamique.

L'une des possibilités consiste à utiliser directement un capteur dit HDR. Quelques technologies sont envisagées. Par exemple, les fabricants de ces capteurs font en sorte de pouvoir maximiser la différenciation du plus petit signal mesurable par rapport au bruit de fond NF ("Noise Floor"). De la même manière, l'augmentation du maximum de signal mesurable FWC ("Full Well Capacity") permet d'augmenter la dynamique du capteur. Spivak et al. [1] répertorient les solutions qui découlent de cette méthode permettant d'élargir la dynamique des capteurs (alors dits "Wide Dynamic Range"). D'autres capteurs HDR

---

1. Pour plus de commodité, l'acronyme "LDR", "HDR" et le mot "dynamique" en remplacement du terme "gamme dynamique" seront utilisés pour la suite de la thèse.

utilisent une réponse logarithmique pour la conversion des photons en électrons [2]. Malheureusement, ces capteurs ont un FPN ("Fixed-Pattern-Noise") trop important, notamment pour de faibles intensités lumineuses capturées. Enfin, il existe d'autres méthodes de fabrication des capteurs HDR, tels que celles utilisées pour les capteurs à variation spatiale d'exposition [3], ou à ajustement de capacité. Ceux-ci ont un haut niveau de complexité à la fabrication des pixels (problème de pixels trop gros par exemple), et présentent un CDS ("Correlated Double Sampling") impossible à réaliser dans ces circonstances, la correction de l'offset FPN devient alors limitée. Dans tous les cas, la plupart de ces capteurs sont sur-mesure et fournissent des unités de traitement de pixels spécialement dédiées au prolongement de gamme dynamique [4–9]. Nous ne détaillerons pas ces technologies dans la thèse.

La seconde méthode d'obtention d'image à grande dynamique est celle traitée au travers de cette thèse : la technique de multiple exposition. Il est courant d'utiliser un capteur conventionnel de type LDR ("Low Dynamic Range") à 8, 10 ou 12 bits par pixel en sortie du capteur, afin de capturer toute l'information disponible d'une scène à grande dynamique. La technique consiste à faire varier les temps d'exposition des images acquises successivement. En limitant le temps d'exposition, on privilégie la capture des détails bien visibles dans les zones fortement éclairées. À contrario, augmenter le temps d'exposition lors d'une capture permet de mieux récupérer l'information présente dans les zones faiblement éclairées. Un exemple de capture d'une scène à grande dynamique est montré sur la Figure 1.2. Les algorithmes d'imagerie complexes permettent de construire une seule image HDR (appelée aussi carte de radiance), qui couvre toute la dynamique des luminosités représentatives de la scène, en combinant les informations pertinentes des acquisitions successives. Un exemple d'image HDR obtenue à partir des cinq images LDR de la Figure 1.2 est montré dans la Figure 1.3.

La technique d'exposition multiple a reçu beaucoup d'attention ces dernières années. La radiance<sup>2</sup> du monde réel est enregistrée par le capteur, et convertie en valeur de pixel. Des algorithmes déterminent la cartographie globale de la radiance  $L$  pour des valeurs de pixels  $Z$  à partir d'un ensemble d'images différemment exposées. Debevec et al. [10], Robertson et al. [11] ou Mitsunaga et al. [12] ont développé des algorithmes d'approches différentes afin de déterminer la courbe de réponse d'un capteur standard à partir de plusieurs expositions réalisées à l'aide d'un système photographique (lentille & capteur & ADC).

Dans un second temps, d'autres ont travaillé sur les TMOs ("Tone Mapping Operators"), Drago et al. [13], Reinhard et al. [14] ou bien Durand et al. [15] pour ne citer que les plus populaires. Ils ont pour fonction de compresser la dynamique des images HDR réalisées. Le but final est de pouvoir afficher toute l'information des images HDR sur un écran standard de faible dynamique. Les rendus sont différents, mais ces opérateurs essaient de

---

2. Quantifie la radiation reçue ou émise dans une direction donnée. Ce terme sera défini plus précisément dans la Section 2.2.

faire en sorte de reproduire le plus fidèlement possible la sensation visuelle de la scène réelle à l'écran. Chaque algorithme TMO a ses propres avantages et inconvénients. De nombreuses études s'attardent sur l'évaluation objective ou subjective de la qualité de ces opérateurs. L'approche conseillée ici est avant tout de sélectionner un TMO suivant les besoins de son application, plutôt que de se référer aux études de qualité souvent contradictoires et non exhaustives.

Une étude approfondie de la littérature, à la fois sur la synthèse HDR et sur les techniques de tone mapping, souligne le fait que la plupart des systèmes mis en œuvre jusqu'à aujourd'hui sont principalement réalisées sur des architectures logicielles, s'affranchissant ainsi des contraintes matérielles et de timing. De plus, la plupart des travaux traitent des images HDR fixes, et peu de publications s'intéressent au domaine de la vidéo HDR. Les architectures logicielles nécessitent souvent plusieurs secondes de calcul lors de la construction d'une simple image HDR. Il est intéressant de pouvoir produire un système qui permette d'augmenter la dynamique de capture, et ce par l'intermédiaire d'une vidéo en temps réel directement visible, en l'intégrant dans une caméra intelligente.

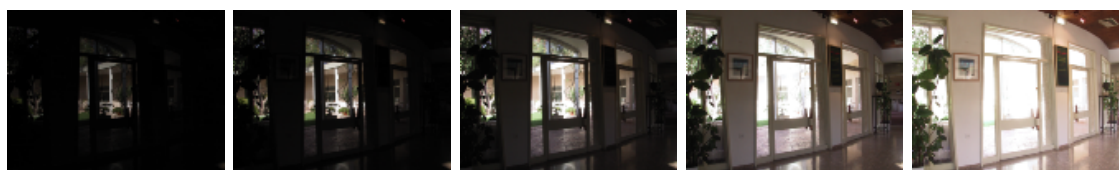
Le travail décrit dans cette thèse a pour but de développer un système complet, basé sur une architecture matérielle, capable d'acquérir, de générer et d'afficher du contenu à grande dynamique, à la cadence du capteur et sans latence apparente pour le spectateur. Les méthodes actuelles basées sur la vidéo HDR sont encore trop rares et trop coûteuses en termes de ressources matérielles [16] pour être efficaces et exploitables directement.



**FIGURE 1.1** – Analogie entre photographie classique et perception visuelle humaine du point de vue du peintre. Illustration des limites de la photographie numérique au travers d'une image résultante d'une capture classique (a), et d'une peinture de Pierre Bonnard (b).

## 1.2/ CONTEXTE DE LA THÈSE

Les travaux ont été effectués dans l'équipe "Architecture et capteurs d'images" du Laboratoire Électronique Informatique Image (LE2I UMR 6306) de l'Université de Bourgogne



**FIGURE 1.2** – Exemple d'une série d'images prises avec des temps d'exposition différents, technique aussi appelée "bracketing". Les images sont tirées du travail de Fattal et al. [17].



**FIGURE 1.3** – Image HDR après tone mapping, réalisée à partir de la séquence d'images de la Figure 1.2.

à Dijon. Les axes de recherches de cette équipe sont identifiés dans le domaine de l'adéquation algorithme-architecture, et concernent les systèmes d'acquisition d'images, les architectures matérielles et les applications temps réel de traitement d'images à fortes contraintes. L'objectif majeur est la validation des concepts théoriques par la réalisation de démonstrateurs à base de systèmes électroniques et informatiques. Les recherches proposées dans le cadre de cette thèse, font suite à des travaux menés dans l'équipe sur la conception de systèmes de vision à haute performance [18–20]. Ils s'intègrent plus spécifiquement dans le cadre du projet européen High Dynamic Range - Low Noise CMOS imagers (HIDRALON 2009-2012) dédié au développement de nouvelles approches de conception de capteurs d'images. L'objectif est de concevoir et mettre au point de nouvelles méthodes permettant d'accroître de manière drastique la dynamique des capteurs d'images tout en conservant, voire améliorant les autres caractéristiques importantes que sont la résolution, la qualité d'image, la sensibilité, etc. Les travaux du LE2I dans ce projet concernent principalement le traitement en temps réel des données issues du capteur.

Certains composants, matériel et IPs ("Intellectual Property") utilisés pour le prototypage dans cette thèse, sont issus d'une collaboration avec les entreprises et services présentés dans le Tableau 1.1.

## 1.3/ BUT DE LA RECHERCHE ET PROBLÉMATIQUE

### 1- Étudier les algorithmes HDR

Collaboration avec les entreprises et les services au cours de la thèse.	
<i>Nom</i>	<i>Apports pour la thèse</i>
Université de Bourgogne	Design de cartes caméra, support objectif, bague allongée, soudure, prêt de matériel et outils de prototypage (Suite Orcad)
E2v	Entreprise participante au projet Hidralon. "Demo Kit" fourni, échantillons de capteurs noir et blanc et couleur fournis en version Beta (Sapphire - EV76C560 ABT/ACT), SAV, conseil
Thales Angénieux	Entreprise participante au projet Hidralon. Développement IP de contrôleur DVI, partenariat lors du test du capteur E2v

**TABLE 1.1** – Collaboration avec les entreprises et les services au cours de la thèse.

Une bibliographie sur le domaine HDR a été réalisée, parcourant les articles qui traitent de la capture, de la génération HDR, du tone mapping, en passant par l'évaluation de la qualité des algorithmes et aux tentatives de définition de systèmes globaux temps réel ou non. Plus de 150 publications datant de 1995 à 2013, telles que des revues indexées, des conférences internationales, des thèses et des livres ont été répertoriés au cours de la thèse. Des algorithmes issus de cet état de l'art ont été testés au cours de la thèse grâce à l'utilisation de Matlab et de logiciels HDR existants tels que Photomatix [21] ou Enfuse [22].

## 2- Générer des images HDR

Nous avons développé plusieurs environnements software et hardware de test et de simulation, nous permettant d'expérimenter les algorithmes HDR. Grâce aux nombreux algorithmes disponibles en code source Matlab libre, un logiciel nommé *HDR-Lab* a été réalisé, dans le but de déployer des algorithmes dans une interface simple et conviviale. HDR-Lab permet la génération HDR, la comparaison entre algorithmes, l'analyse de qualité, l'analyse statistique, la visualisation des différents résultats, etc. De la même manière, une fois sélectionnés, certains algorithmes HDR ont été décrits dans un langage de niveau plus bas, dans le but de mieux préparer les algorithmes à une future implémentation matérielle. De plus, la description des algorithmes en un langage très bas niveau de type VHDL, permet de mieux déboguer l'application. De même, nous avons réalisé un environnement de simulation des algorithmes, au travers d'une simulation matérielle sous Modelsim, mettant en œuvre une simulation de capteur, un pipeline de création HDR+TMO, et un système de visualisation du résultat au format PGM ("Portable Gray-Map").

## 3- Définir un pipeline HDR adapté aux contraintes hardware

Le problème soulevé par cette thèse n'est pas la difficulté de générer des images HDR à proprement parlé, mais d'inclure cette fonctionnalité dans un procédé direct de vision

en temps réel. Le capteur nous envoie une série d'images que nous devons traiter dans le flux, afin de ressortir du contenu HDR directement visible pour le spectateur. De nombreux systèmes dédiés HDR, de la capture à l'affichage, en passant par la génération HDR et le tone mapping ont été inventés. Seulement, très peu d'entre eux traitent d'un pipeline HDR complet (capture, création HDR, tone mapping et visualisation). En outre, notre but est de faire en sorte que notre pipeline associé du capteur LDR, puisse être considéré comme un capteur HDR à part entière, puisqu'il fournit du contenu HDR à la même cadence qu'un capteur de vision standard. Grâce aux études faites sur les algorithmes HDR, à leur implantation sous Matlab et C++, nous connaissons leur complexité et leur faisabilité en termes d'implémentation matérielle. Le choix des algorithmes sera fait en fonction de nos ressources, tout en se préoccupant aussi de la bonne qualité du rendu. Une adéquation algorithme-architecture sera donc au cœur de nos préoccupations pour ce projet.

#### *4- Développer un prototype de vision HDR*

Le développement d'une caméra vidéo HDR est la principale contribution qu'apporte la thèse. Plusieurs étapes ont été nécessaires à l'élaboration d'une telle caméra. Les algorithmes et techniques étudiés ont permis de se fixer des contraintes. Ces contraintes peuvent aller de la complexité des algorithmes mis en œuvre, du choix de la plateforme et des outils de développement, de la mémoire embarquée, ou encore de la configuration du capteur. L'architecture s'est basée autour du capteur d'image développé et fournit gracieusement en version Beta (Sapphire - EV76C560 ABT/ACT) par e2v, dans le cadre du projet Européen Hidralon. Ce capteur est un capteur LDR conventionnel, "low power" et pourvu de quelques fonctionnalités avantageuses. Il a donc été jugé adapté au développement d'une architecture embarquée de type smart caméra. Des cartes servant de support au capteur (alimentation+routage des bus de données) ont été construites en partenariat avec l'université. Ces cartes sont destinées à être branchées sur les plateformes d'évaluation à base de FPGA.

## 1.4/ STRUCTURE DU MANUSCRIT

Le CHAPITRE 2 présentera les notions fondamentales en relation avec le domaine HDR ; de la définition de la radiométrie aux concepts ayant donné naissance au High Dynamic Range, tels que la gamme dynamique ou les limites du capteur d'image. Nous énoncerons les principaux algorithmes dédiés HDR, pour terminer par les problèmes rencontrés lors de l'application de ces méthodes dans un contexte axé purement vidéo.

Le CHAPITRE 3 aura pour cible la vision HDR, s'articulant autour de la description des principaux systèmes matériels existants, de leurs fonctionnalités et de leurs lacunes. Cet état de l'art aura pour but de nous aider à définir un choix d'architecture matérielle propice à l'implémentation d'un pipeline HDR pour une vidéo temps réel.

Le CHAPITRE 4 fera l'objet de la description complète de notre caméra HDR. Nous nous attarderons sur les capacités algorithmiques et fonctionnelles développées dans le but de réaliser notre caméra HDR : la caméra *HDR-ARtiSt*. Ces aspects seront traités au travers de ces quelques points :

- L'élaboration d'un algorithme d'auto exposition dédié HDR
- La description d'un nouveau système de gestion de mémoire et de bufferisation des images
- La création d'un logiciel de test des algorithmes HDR : *HDR-Lab*
- La sélection des algorithmes HDR et leurs simplifications envisagées.

Le CHAPITRE 5 fera la description complète de la mise en œuvre du système décrit dans le CHAPITRE 4 sur une plate forme matérielle. Des résultats de simulations software, hardware et des comparaisons faites avec quelques systèmes existants seront présentés. Par ailleurs, nous montrerons les résultats finals de notre caméra après implémentation hardware, par l'intermédiaire de vidéos de démonstration disponibles sur support annexe et/ou sur internet.

Nous concluons par le CHAPITRE 6, où nous donnerons un bilan de nos contributions, et où nous envisagerons les traditionnels "Futur Works" et perspectives possibles de nos travaux.





# ALGORITHMIE POUR L'IMAGERIE HDR

## 2.1/ INTRODUCTION

Les domaines de l'imagerie médicale, la télévision, la vidéosurveillance, la réalité virtuelle ou les applications militaires ont besoin de retranscrire des images de la manière la plus fidèle possible. C'est un challenge particulièrement difficile. En effet, retransmettre la réalité telle que le perçoit l'œil humain nous met en face de contraintes matérielles qui sont à résoudre. Les scènes réelles ont souvent une échelle de dynamique de luminance de  $10^9 : 1$ , par exemple dans le cas d'une scène extérieure où se mêlent les zones très sombres et les zones très fortement éclairées (un sous-bois par exemple). L'œil humain est un organe composé de photorécepteurs qui convertissent la lumière en signaux neurochimiques. Les cônes et les bâtonnets sont les deux types de photorécepteurs. Les bâtonnets sont plus sensibles que les cônes, mais ne permettent pas la distinction des couleurs. Voilà pourquoi dans les zones sombres, nous ne percevons pas bien les informations relatives à la couleur. L'imagerie à grande dynamique est capable de manipuler physiquement les valeurs de luminances réelles de la lumière visible, qui sont pour la plupart comprises entre  $10^{-3} \text{cd.m}^{-2}$  et  $10^6 \text{cd.m}^{-2}$ .

Les capteurs d'imagerie conventionnels sont capables d'avoir en sortie une gamme dynamique de luminance de  $8 \sim 14 \text{bits}$ . De la même manière, les écrans standards ne peuvent afficher qu'une gamme limitée de dynamique  $100 \sim 1000 : 1$ . Les méthodes d'imagerie traditionnelles n'utilisent pas les valeurs physiques de luminances, et sont limitées par la technologie qui ne manipule que 8 bits par canaux RGB en termes de profondeur de représentation des données. Pour être clair, la capture propre des luminances (ou radiances) d'un environnement demande une meilleure précision qu'une simple représentation RGB 24 bits. Le calcul de ces vraies valeurs physiques se fait via l'utilisation d'un format capable de représenter le spectre visible complet. Ce même format est utilisé pour l'imagerie à grande dynamique, où les diverses informations issues des différentes images capturées sont assemblées et sauvegardées dans une carte de radiance, ou image HDR.

Reinhard et al. [23] ont fait un examen exhaustif des différents domaines d'applications des caméras HDRs. Par exemple, il est évident que la technique HDR est un choix plus

que stratégique dans le cas de la photographie en astronomie. En effet, les scènes capturées du cosmos par les télescopes couvrent une immense gamme de luminosité, allant de la représentation des simples nébuleuses ou galaxies lointaines, aux éclats incandescents du soleil. Malheureusement les désavantages intrinsèques au HDR, tels que les problèmes d'alignement ou de pondération des pixels à fusionner, sont exacerbés lorsque l'on l'applique le HDR à l'astronomie. Des temps d'exposition pouvant parfois dépasser la dizaine de minutes, des vibrations dû à l'obturateur, le vent ou une distance focale très grande, sont des paramètres à prendre sérieusement en considération. Le HDR est aussi essentiel pour les applications de sécurité en plein air pour lequel un changement d'illumination imprévisible peut affecter la performance des algorithmes [24–26]. De même, les techniques HDRs devraient faciliter le suivi d'objet [27] ou les applications automobiles [28, 29] sous une illumination non maîtrisée. Les récents travaux de recherche portant sur la vision artificielle ont clairement démontré les avantages du temps réel pour la vision HDR [30, 31]. Enfin, les applications médicales nécessitent des images de haute précision, et des techniques d'éclairage HDR de pointe, afin d'améliorer la qualité du rendu des systèmes d'imagerie [32, 33]. Cela est particulièrement vrai pour la biologie [34], les systèmes d'imagerie basés sur la fluorescence [35, 36], ou les endoscopes [37, 38].

Ainsi, la lumière du monde réel peut maintenant être capturée, quantifiée, stockée et utilisée pour de nombreuses applications (voir exemples en Figure 2.2). Une autre application intéressante est le ré-éclairage dynamique par fusion d'images LDRs, permettant ainsi de contrôler indépendamment les réglages de chaque source de lumière présente dans une scène. Le logiciel HDR ReLight permet par exemple d'exploiter cette fonctionnalité, avec une interface logicielle permettant d'allumer ou d'éteindre artificiellement les sources de lumière dans l'image. Les zones très sombres et très éclairées d'une scène peuvent être enregistrées une à une dans une image ou une vidéo, supprimant ainsi les zones sur exposées et sous exposées.

Il existe une caméra commerciale, capable de saisir à la fois des informations présentes sur des objets incandescents, et sur des ombres infimes. Cette caméra se nomme la *SpheronCam HDR*. Elle est capable de récupérer 26 stops<sup>1</sup> de dynamique de luminosité réelles, sur une image de 50 Mégapixels en 360°. L'application est en fait dédiée à la mesure de la vraie luminosité d'un endroit, et à l'utilisation de celle-ci pour éclairer les objets par une technique d'infographie, suivant le procédé "Image-based Lighting".

Par extension, tous les domaines où il y a un traitement d'image sur des fichiers numériques LDRs, pourraient aussi bien supporter nativement la technique HDR à l'avenir. On ne ferait donc plus les calculs sur une dynamique simple de 8-bit par canal, mais sur une dynamique beaucoup plus élevée.

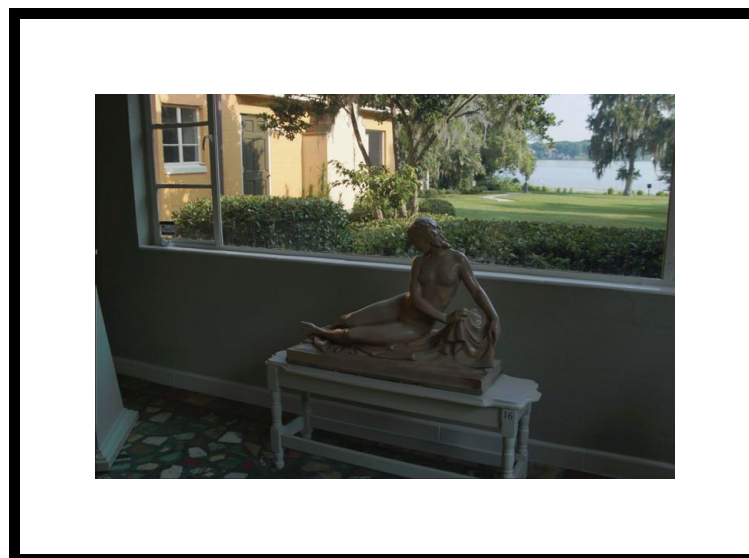
Une des méthodes HDR est la multiple acquisition, où les images prises avec des temps

---

1. Un stop supplémentaire signifie doubler la quantité de lumière de la prise de vue

d'exposition différents sont assemblées pour former l'image HDR. Il existe plusieurs techniques de fusion des images, dont celle de Debevec et Malik [10]. L'image HDR résultante est codée sur un grand nombre de bits. La gamme dynamique des images HDR ne peut pas être représentée directement sur des écrans de dynamique plus faible. Les écrans dits "HDR", pouvant afficher une plus grande dynamique de luminances que les écrans standards, sont en phase de développement. Ils sont effectivement encore trop rares et commencent à peine à arriver à commercialisation. Il faut donc pour l'instant trouver un autre moyen d'afficher les images HDRs. Ce moyen est la compression de dynamique, avec l'aide d'une technique que nous appelons le *tone mapping*.

Dans ce chapitre, nous nous pencherons sur les définitions des termes spécifiques au domaine à grande dynamique, tels que la radiance ou l'illuminance, qu'est ce que la gamme dynamique, etc. Nous nous attarderons par la suite sur la description des capteurs LDR conventionnels, en identifiant leurs lacunes en termes de dynamique, pour ensuite envisager des solutions à ces problèmes. Nous concluons en détaillant la méthode la plus utilisée à ce jour pour augmenter la dynamique des capteurs : la méthode de capture d'expositions multiples.

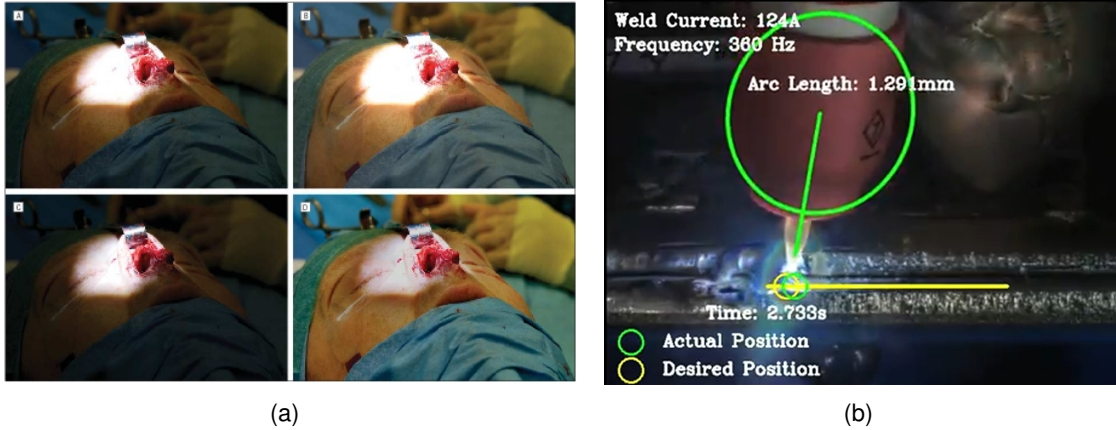


**FIGURE 2.1** – Exemple d'image HDR issue du livre de Debevec [23]. On remarque que l'on peut distinguer à la fois les hautes luminosités provenant de l'extérieur, et les basses luminosités de l'intérieur de la pièce.

## 2.2/ DÉFINITIONS

### 2.2.1/ GRANDEURS RADIOMÉTRIQUES ET PHOTOMÉTRIQUES

Nous définissons ici les grandeurs physiques avec lesquelles travaillent les chercheurs dans le domaine HDR. L'image de la Figure 2.1 montre comment on représente une



**FIGURE 2.2** – Applications de l'imagerie HDR. Une opération chirurgicale (a) et un guidage de soudure (b).

scène réelle à grande gamme dynamique au travers d'une simple image 8-bit numérisée.

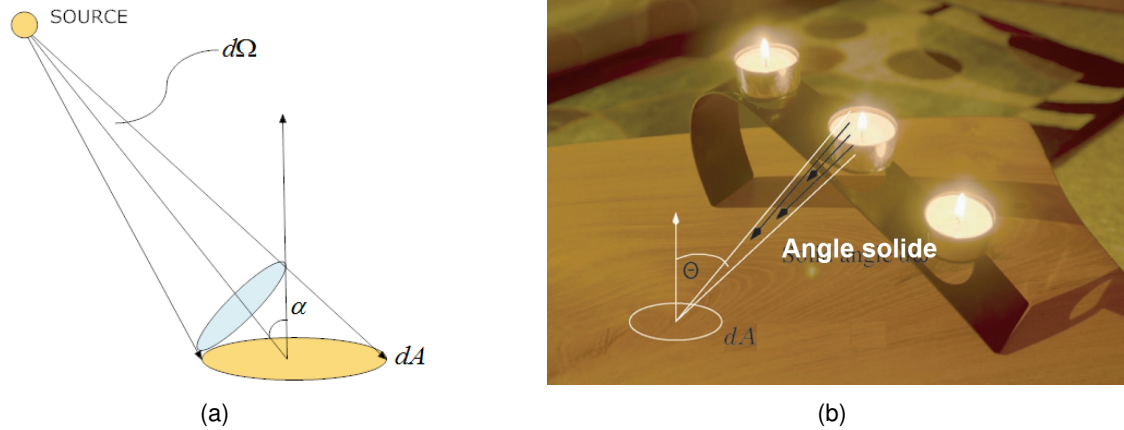
### 2.2.1.1/ LA RADIOMÉTRIE

La radiométrie est la science de la mesure physique du rayonnement électromagnétique (lumineux dans notre cas). La lumière est une onde électromagnétique dont l'énergie est mesurée en *Joules* et notée ici  $Q_e$ . La lumière a la capacité de se propager dans des milieux tels que l'air, l'eau ou l'espace. Nous pouvons dès lors, dériver cette quantité pour en quantifier et qualifier sa propagation. Par exemple, la mesure du flux par unité de temps, définit la puissance électromagnétique (ou flux électromagnétique), telle que  $\Phi_e = \frac{dQ_e}{dt}$ . Si l'on mesure cette puissance par unité d'angle solide, on définit alors l'intensité électromagnétique,  $I_e = \frac{d\Phi_e}{d\Omega}$ , où  $\Omega$  est l'angle solide en stéradian. La luminance énergétique (ou radiance)  $L_e$  indique la puissance électromagnétique émise/reçue par une surface émettrice/réfléctive dans une direction particulière. On la mesure en Watt par stéradian par mètre carré ( $W.sr^{-1}.m^{-2}$ ). La radiance  $L_e$  s'exprime par :

$$L_e = \frac{d^2\Phi_e}{dA d\Omega \cos \theta}, \quad (2.1)$$

où  $\Phi_e$  est la puissance du flux électromagnétique ( $W$ ),  $A$  est la surface apparente ( $m^2$ ),  $\Omega$  est l'angle solide et  $\theta$  est l'angle que fait la perpendiculaire à la surface de vision avec la direction. Un schéma illustre le procédé sur la Figure 2.3.

Lorsque l'on focalise la puissance électromagnétique sur une surface avec une lentille par exemple, on parle d'éclairement énergétique (ou irradiance notée  $E_e$  en  $W/m^2$ ). L'irradiance d'un point de l'image optique est proportionnelle à la radiance du point correspondant de la scène. La relation linéaire entre radiance et irradiance dans le cas de lentilles



**FIGURE 2.3** – Illustration de la mesure de radiance ou de luminance. Elle représente le flux de puissance électromagnétique/lumineuse par unité de surface, en provenance ou dans une direction donnée. L'illustration (b) est tirée du livre de Reinhard et al. [23]

minces est donnée par la formule suivante :

$$E = L \frac{\pi}{4} \left( \frac{d}{f} \right)^2 \cos \alpha^4, \quad (2.2)$$

où  $d$  est le diamètre de la lentille,  $f$  est la distance focale de la lentille, et  $\alpha$  est l'angle que fait l'axe optique de la lentille avec le faisceau de lumière incident. La construction des lentilles actuelles corrigent en grande partie l'influence de l'angle  $\alpha$  sur la proportionnalité entre la radiance et l'irradiance. Les rayons divergents d'un point de la scène sont donc rassemblés et concentrés sur le capteur d'image. Pour résumer, la conversion de radiance vers l'irradiance est déterminée par les propriétés de l'optique.

### 2.2.1.2/ LA PHOTOMÉTRIE

Le système visuel humain est capable de percevoir des rayonnements dont la longueur d'onde  $\lambda$  est comprise entre 380 et 780 nanomètres. De plus, l'œil a une sensibilité  $V(\lambda)$  différente pour chaque longueur d'onde de la gamme du visible. La photométrie est la mesure de la lumière du point de vue de l'œil humain. On peut l'assimiler à la radiométrie en ayant une pondération spectrale supplémentaire, correspondant à la réponse spectrale de l'œil. Les outils de mesures photométriques sont étalonnés grâce à cette caractéristique. Il existe donc une analogie entre les deux domaines. On parle alors non plus d'irradiance ni de radiance, mais d'illuminance (ou éclairement lumineux) et de luminance, lorsque l'on cherche à mesurer une quantité lumineuse dans une direction donnée du point de vue de l'œil humain. On la mesure en candela par mètre carré ( $cd.m^{-2}$ ). Voici la fonction qui fait la correspondance entre la radiance  $L_{e,\lambda}$  pour une longueur d'onde, et la luminance  $L_v$  :

$$L_v = \int_{780}^{380} L_{e,\lambda} V(\lambda) d\lambda \quad (2.3)$$

Les autres correspondances entre les deux domaines, ainsi que les unités et notations des différentes grandeurs photométriques et radiométriques sont détaillées dans le Tableau 2.1.

Lorsqu'une image est acquise, la lumière est focalisée par une lentille, qui limite le nombre de directions à partir desquelles la lumière est reçue. Le capteur d'image est divisé en petites portions appelées pixels, de sorte que chaque pixel reçoit la lumière sur une petite surface. Il est possible de relier les tensions extraites d'un capteur à la mesure de radiance, si les pixels correspondants ne sont ni sous-exposés, ni surexposés du point de vue du capteur.

La lumière qui frappe un point d'une surface depuis une direction particulière constitue la base de l'information d'une image. Toutefois, quand nous photographions une scène, que ce soit avec un film ou un système électronique, la numérisation bidimensionnelle de l'ensemble des luminosités est rarement représentative des mesures réelles de rayonnement électromagnétique. En effet, le système optique est constitué d'une ouverture, d'une lentille, d'un capteur, de convertisseurs ADCs ("Analog to Digital Converters") qui contraignent l'arrivée de la lumière et qui la transforment de façon non linéaire. Par exemple, si la valeur d'un pixel  $Z_1$  est deux fois supérieure à un pixel  $Z_2$ , il est peu probable que le rayonnement  $E_1$  retranscrit par le pixel  $Z_1$  soit deux fois supérieur à  $E_2$ . L'estimation de la radiance réelle est donc difficile à réaliser dans le contexte non linéaire du processus photographique.

Dans toute la thèse, nous utiliserons les termes de radiance et d'irradiance au lieu des termes français de luminance énergétique et d'éclairement énergétique, pour parler de la quantité capturée par la caméra. Dans la littérature concernant le tone mapping, certains articles intègrent la lumière dans le domaine photométrique, et d'autres dans le domaine radiométrique. Il en résulte un mélange entre les termes de radiance, luminance, intensité, illumination ; ce qui peut entraîner de la confusion. Dans la thèse, nous utiliseront le terme de luminance pour signifier la mesure de l'éclairement de la scène perçue par le spectateur.

Nous n'envisagerons pas de traiter en détails les problèmes HDR liés aux aspects colorimétriques dans cette thèse, les caméras construites et utilisées au cours de celle-ci étant principalement conçues pour le noir et blanc.

## 2.2.2/ NOTATIONS

La liste des notations utilisées tout au long de cette thèse est représentée dans le Tableau 2.2.

Les grandeurs radiométriques et photométriques.		
Grandeur	Unité	Notation
<b>Grandeur radiométrique</b>		
Énergie électromagnétique	<i>Joule</i>	$Q_e$
Puissance électromagnétique	$W (J.s^{-1})$	$\Phi_e$
Intensité électromagnétique	$W.sr^{-1}$	$I_e$
Radiance	$W.sr^{-1}.m^{-2}$	$L_e$
Irradiance	$W.m^{-2}$	$E_e$
<b>Grandeur photométrique</b>		
Quantité de lumière	<i>lm.s</i>	$Q_v$
Puissance lumineuse	<i>lumen</i>	$\Phi_v$
Intensité lumineuse	<i>candela</i>	$I_v$
Luminance	$cd.m^{-2}$	$L_v$
Illuminance	$lux (lm.m^{-2})$	$E_v$

TABLE 2.1 – Les grandeurs radiométriques et photométriques.

### 2.2.3/ LA GAMME DYNAMIQUE

La gamme dynamique est la différence entre la plus grande valeur et la plus petite valeur mesurable d'une quantité. Celle-ci peut s'exprimer de plusieurs façons, suivant le domaine dans lequel elle est référencée. En photographie, on peut la mesurer en nombre de "stops", lorsque l'on fait référence à une gamme dynamique pendant la capture d'une scène. On peut aussi la mesurer en Décibels (dB), si l'on fait référence à la dynamique du capteur d'image. Pour les afficheurs, la différence de contraste entre les pixels affichables les plus sombres et les plus lumineux, s'exprime de la manière suivante :

– 1 : 100, par exemple, pour un écran CRT (Cathode Ray Tube).

Plus généralement, une formule donnant la dynamique peut être écrite ainsi :

$$Dr = 20.\log_{10}\left(\frac{X_{max}}{X_{min}}\right), \quad (2.4)$$

où  $X_{min}$  et  $X_{max}$  sont la plus petite et la plus grande valeur de quantité mesurable.

Lorsque l'on pratique la photographie, la valeur d'exposition (notée  $EV$ ) est un indicateur des paramètres d'une caméra, permettant de quantifier un intervalle sur une échelle d'exposition. Cette valeur d'exposition se réfère à une certaine combinaison d'ouverture (f-number) et de temps d'exposition ( $\Delta t$ ). La différence entre 2 ouvertures et 2 temps d'exposition voisins est alors appelée un *stop*. Les appareils modernes ont souvent des valeurs d'exposition  $EV$  graduées en "demi-stop" ou "tiers-stop". Pour plus de commodité, on utilise le logarithme de base 2 pour travailler avec ces plages dynamiques. En effet, lors de la capture, un stop en plus ou en moins double ou divise par deux la quantité de lumière acquise. La formule qui permet de remonter à la valeur d'exposition à partir de



Liste des notations fréquemment utilisées dans cette thèse.	
Notation	Désignation
$D_y$	Plage dynamique
$P$	Nombre d'images utilisées pour la reconstruction d'une image HDR
$I_p$	$p$ indexe l'image $I$ dans la séquence de $P$ images
$I_{L/M/H}$	Indexation des images d'exposition basse ( $L$ ), moyenne ( $M$ ) et haute ( $H$ )
$M, m$	Nombre de lignes d'une image et son indexation
$L_m$	Indexation d'une ligne
$N, n$	Nombre de colonnes d'une image et son indexation
$i$	Indexe la position du pixel dans une ligne
$j$	Indexe la position du pixel dans une colonne
$Z_{ij}^p$	Valeur du pixel à la position $(i, j)$ dans la $p^{ieme}$ image
$E_{ij}$	Luminance (ou radiance en radiométrie) du pixel HDR à la position $(i, j)$
$f$	Courbe de réponse de la caméra ( $g = \ln f^{-1}$ )
$\Delta t_p$	Temps d'exposition de la $p^{ieme}$ image
$D_{ij}$	Pixel affichable (post-tone mapping) à la position $ij$
$D_{min}, D_{max}$	Valeurs minimum et maximum de la dynamique de l'image après tone mapping. En général $D_{min} = 0$ et $D_{max} = 255$ pour des images codées sur 8 bits
$W \& R$	Opérations d'écriture et de lecture dans la mémoire

TABLE 2.2 – Liste des notations fréquemment utilisées dans cette thèse.

l'ouverture et du temps d'exposition est la suivante :

$$EV = \log_2 \left( \frac{\text{f-number}}{\Delta t} \right). \quad (2.5)$$

Dans le domaine HDR, on cherche en priorité à mesurer et capturer la gamme dynamique complète d'une scène, c'est à dire à quantifier et enregistrer toutes les valeurs de luminances. La différence de luminance entre les éléments les plus éclairés et les plus sombres d'une scène est ainsi considérée. L'approche diverge quelque peu par rapport à la photographie conventionnelle, puisque l'on raisonne en termes de quantité physique de lumière. Dans la plage des luminances, on travaille avec le domaine logarithmique de base 10.

Pour un capteur d'image, le  $SNR$  ("Signal to Noise Ratio") permet de quantifier la différence entre la plus grande quantité de lumière qui sature le capteur, et la plus petite valeur discernable immédiatement au dessus du niveau de bruit. On utilise alors le logarithme de base 10 pour plus de commodité.

Pour résumer, voici les différentes façons de mesurer la plage dynamique en fonction du contexte :

1. Rapport de contraste - pour les afficheurs -  $Dr_C = \frac{X_{max}}{X_{min}}$
2. Rapport d'exposition - pour la photographie -  $Dr_{stop} = \log_2 \left( \frac{X_{max}}{X_{min}} \right)$ , mesurée en f-stops

3. Rapport signal sur bruit ( $SNR$ ) - pour les capteurs d'images  $Dr_{SNR} = 20 \cdot \log_{10}(\frac{X_{max}}{X_{min}})$ , mesurée en  $dB$

En français et en photographie, on peut emprunter le terme de "dynamique" pour la simplification du terme "gamme dynamique". En anglais, le terme utilisé est dans tous les cas "dynamic range". Dans la thèse, afin d'éviter la lourdeur, nous utiliserons le terme "dynamique" seul pour parler d'une plage dynamique de luminances.

<i>Contrastes</i>	<i>Decibels (dB)</i>	<i>Stops</i>
1	0	0
2	6.02	1
3.16	10	1.66
4	12.04	2
5	13.98	2.32
8	18.06	3
10	20	3.32
16	24.0	4
20	26.0	4.32
31.6	30	4.98
32	30.2	5
50	34.0	5.64
100	40	6.64
1 000	60	9.97
1 024	60.2	10
10 000	80	13.3
100 000	100	16.6
1 000 000	120	19.9
100 000 000	160	26.6
10 000 000 000	200	33.2

**TABLE 2.3** – Valeurs et correspondances de dynamiques en gammes de contrastes, decibels et stops.

#### 2.2.4/ LES LIMITES DU CAPTEUR D'IMAGE

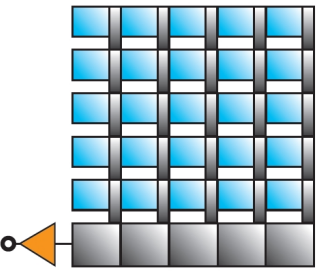
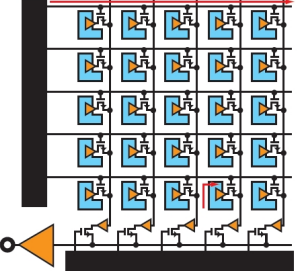
Une caméra est essentiellement un dispositif imparfait pour la mesure de la radiance d'une scène, car elle ne peut pas capturer fidèlement la dynamique complète et son contenu spectral. Le film ou capteur d'image dans une caméra numérique classique est exposé à la lumière issue du système optique. La lentille est un élément passif qui recentre la lumière incidente sur le plan du capteur d'image. Toute l'information est bien présente (exceptées les quelques réflexions créées par le système optique), mais des limites dans la conception des capteurs sont bel et bien présentes.

Les capteurs APS ("Active Pixel Sensor"), produits à partir de la technologie CMOS, ont succédé à la technologie CCD (Charge-Coupled Device) dans les années 90 [39].

Le capteur CCD est devenu la principale technologie utilisée dans les capteurs d'image scientifiques à partir de 1970. La technologie CCD est très complexe, nous ne la détaillerons pas ici. La sensibilité élevée résulte d'un rendement quantique de l'ordre de 40%, et de la présence d'amplificateurs à faible bruit. Le bruit typique de l'amplificateur de sortie est de l'ordre de 5 électrons. Même si le CCD est devenu la technologie des détecteurs de base pour l'imagerie visible, cette technologie a atteint ses limites. Par exemple, les capteurs CCD sont plus sensibles aux effets de blooming (ou éblouissement) et de smearing (traînées verticales en cas de surexposition), à cause notamment de la grande sérialisation des données. Ces lacunes ont conduit au développement de nouvelles technologies d'imagerie alternatives telles que le CMOS. Un récent "white paper" de chez *Grass Valley*, une entreprise fabricant des caméras pour la télévision, a annoncé dernièrement que *"La performance des capteurs CMOS des caméras de télédiffusion s'est grandement améliorée au cours des dernières années et leur développement est tel que les clients de la télédiffusion choisissent la technologie CMOS pour leurs nouvelles caméras"*. En effet, les difficultés rencontrées deviennent nombreuses pour le CCD, telles que la trop grande sensibilité aux radiations, la difficulté de construire de larges matrices de pixels, la consommation excessive, la réponse spectrale limitée, la vitesse de sortie de lecture limitée, etc. De nos jours, la majorité des appareils reflex numériques et les caméras numériques utilisent la technologie CMOS. De plus, cette technologie offre un panel d'évolutions considérable par rapport à la technologie CCD. On peut supposer qu'à l'avenir, toutes les caméras seront équipées d'un ou de plusieurs capteurs CMOS.

Grâce aux progrès dans le domaine de la micro-lithographie, la nouvelle technologie de capteur APS a su préserver les caractéristiques du CCD comme la sensibilité élevée ou des signaux de haute fidélité, tout en corrigeant certains aspects négatifs. Parmi d'autres avantages, l'approche APS ne nécessite pas de transfert de charge sur de grandes distances et élimine ainsi les grands défauts associés aux réseaux de détecteurs CCDs. Un capteur APS est formé d'un ou de plusieurs transistors actifs (jusqu'à 5) qui composent chaque pixel de la matrice d'image. Le transistor actif peut fournir à la fois les fonctions de gain et de mise en mémoire tampon, augmentant ainsi les possibilités de faire directement du prétraitement dit "on chip". En d'autres termes, sur un même circuit, le capteur CMOS peut contenir la partie photosensible mais aussi de l'électronique servant au stockage, à la conversion, à la lecture et au traitement de l'information. Le bruit peut par exemple être corrigé à la source. De plus, l'accès aux pixels dans le capteur CMOS peut se faire de façon aléatoire pour des applications nécessitant la focalisation en ROIs ("Region of Interest"), contrairement à la technologie CCD. Une petite comparaison technologique des deux types de capteurs est faite dans la Figure 2.4.

Néanmoins des problèmes subsistent pour ces deux technologies. La sensibilité limitée lors de la capture de scène où la dynamique de luminance est trop importante, ou le manque de dynamique de codage du pixel de sortie (jusqu'à 12 bits en général), sont les principaux défauts des technologies actuelles. Nous détaillerons ici ces aspects sen-

Particularité	CCD	CMOS
<b>Structure</b>		
<b>Horloge</b>	Grande cadence	Faible cadence
<b>Convertisseur ADC</b>	Externe	Interne
<b>Adressage pixel</b>	Non	Oui
<b>Obturbateur</b>	Global shutter	Rolling shutter et/ou global shutter
<b>Application récente</b>	Télédiffusion	Téléphone portable, cinéma, reflex numérique, télédiffusion
<b>Avantages</b>	Grande sensibilité NIR	Grande sensibilité NIR, faible consommation énergétique, faible production de chaleur, tout numérique
<b>Inconvénients</b>	Forte consommation énergétique, forte production de chaleur, traînées verticales, défauts pixels	Skeed effect pour le rolling shutter, traînées verticales
<b>Dév. attendu</b>	↓	↑↑↑

**TABLE 2.4** – Comparaison CCD VS CMOS. Images tirées de white paper de chez Grass Valley

sibles, d'un point de vue un peu plus technique, sur ce qui conduit au manque de dynamique et à l'élaboration de nouvelles solutions augmentant artificiellement la dynamique de capture :

### 1. La taille de pixel

Plus un pixel peut contenir de signal, plus le potentiel de rapport signal sur bruit (SNR pour "Signal-to-Noise Ratio") est grand, et donc plus la plage dynamique est élevée. Ce paramètre est essentiel pour préserver les détails dans les zones claires et sombres d'une image. Grâce à une surface photosensible plus grosse, on améliore la sensibilité, le facteur de remplissage et la capacité à accueillir le signal. Mais malheureusement ces avantages sont contrebalancés par une baisse du nombre de pixels dans une même surface de silicium, ce qui entraîne un coût de fabrication plus élevé. De même, il y aura une augmentation du bruit (et donc paradoxalement une diminution de la dynamique). De nos jours, un compromis semble avoir été trouvé aux alentours d'une taille de pixel avoisinant les 5/6 microns, ce qui offre

une combinaison optimale en termes de résolution, de faible bruit, de SNR et de coût modéré.

## 2. La sensibilité

La sensibilité en ISO d'un appareil photo influe sur la plage dynamique du système de capture. En effet, lorsque l'on a une sensibilité élevée, le bruit de fond est amplifié et certaines zones sombres sont mal capturées par le détecteur : il y a donc une perte d'information. Néanmoins, dans un contexte vidéo où il y a un désir de capturer les objets en mouvement, il est toutefois nécessaire d'avoir une sensibilité suffisamment grande afin d'éviter qu'un trop grand nombre d'artefacts ne viennent dégrader l'image ou la vidéo.

## 3. Le convertisseur Analogique/Numérique (ADC)

Le signal lumineux directement capturé par le photosite est converti en une valeur analogique, puis en un signal numérique afin d'être acheminé vers la sortie. Cette conversion est réalisée grâce aux convertisseurs ADCs. En général, les capteurs CMOS ont une résolution de 10 à 12 bits pour les ADCs. Le convertisseur est en quelque sorte un des points clés, et son évolution est un enjeu majeur pour la conception des futures caméras et imageurs. Les taux d'échantillonnage du signal analogique sont énormes, pour arriver à capturer en temps réel la lumière et la transmettre sous forme numérique (plusieurs dizaines de MS/s). De plus, si l'on veut augmenter la résolution de ces convertisseurs, on se retrouve face à un problème de taux d'échantillonnage trop élevé, augmentant par là même la consommation et le bruit thermique du système global. De nouvelles technologies ont fait leur apparition à travers les imageurs CMOSs, en intégrant des convertisseurs en parallèle, pouvant ainsi transférer les données hors puce à des taux extrêmement rapides. Les données numériques sont sérialisées dans le but d'obtenir des sorties à grande vitesse, en contrepartie d'une complexité accrue et d'un coût de développement supplémentaire non négligeable.

## 4. Le mode de capture

Il existe deux modes de capture pour les capteurs CMOS : le "global shutter" et le "rolling shutter". De par sa conception, un capteur CCD ne peut être utilisé qu'en mode "global shutter". Dans certains cas, le rolling shutter permet d'avoir une meilleure sensibilité pour le capteur CMOS, lorsque l'on peut effectuer le CDS ("Correlated Double Sampling") dans le but d'éliminer le bruit de reset. Cette méthode d'obturation peut lier temporellement la phase lorsque le capteur fait son acquisition, et la phase où il transfère les données vers la sortie. Ces deux phases ne peuvent pas être associées en mode global shutter, car l'acquisition est stoppée pendant que l'ensemble des données sont transférées. Ceci laisse les photodiodes "au repos" durant le transfert des données. Cependant, un mode dit d'"overlap" a fait son apparition pour le capteur CMOS en mode global shutter à 5 transistors, permettant l'intégration pendant le "readout".

Le mode de capture dit "rolling shutter", est une technique où la capture n'est pas faite simultanément en tout point de la matrice pixel, mais plutôt réalisée par balayage horizontal ligne par ligne, ce qui ne laisse qu'une seule ligne obturée pendant le transfert, pendant que les autres lignes peuvent continuer l'intégration. Au final, les photodiodes sont théoriquement exposées plus longtemps. Le mode global shutter est largement utilisé dans le cas d'une capture vidéo. Jusqu'à ce que la technologie CMOS ne dispose d'un global shutter, il a été difficile d'utiliser le rolling shutter, source d'un trop grand nombre d'artefacts ("jello effect") durant la capture d'une vidéo en présence de mouvement.

## 5. Conclusion

Pour les raisons énoncées précédemment, il paraît difficile pour le moment d'augmenter significativement et intrinsèquement la plage dynamique des capteurs actuels, même sous une technologie CMOS. Certes de tels systèmes suffisent dans le cas d'une simple capture dans un environnement doté d'une dynamique de luminance raisonnable. Mais pour certaines scènes où les conditions de luminosité sont difficiles (fortes différences d'éclairage local ou changement brutal de luminosité), les réelles lacunes des capteurs deviennent assez handicapantes. Ces situations sont souvent la cause de dysfonctionnements ou d'interprétations faussées pour les applications de vidéo surveillance par exemple. La méthode HDR permet en partie de surmonter ce cas de figure.

## 2.3/ CRÉATION D'IMAGE HDR À PARTIR D'EXPOSITIONS MULTIPLES

Nous avons choisi la technique de capture à multiples expositions pour créer les images HDR. Les différentes techniques sont présentées dans la Figure 2.4. Une fois un set d'images sélectionné, nous avons le choix de les fusionner directement dans le domaine image 8-bit, ou bien de passer dans le domaine de radiance, représentatif des vraies valeurs de luminosité du monde qui nous entoure. Nous présenterons les deux méthodes de fusion, pour n'en choisir qu'une à implémenter dans notre futur système de vision HDR.

### 2.3.1/ FUSION D'EXPOSITIONS DANS LE DOMAINE DE RADIANCE

La récupération de la carte de radiance se fait en deux étapes :

1. L'estimation de la courbe de réponse inverse du système optique
2. La transformation des pixels de toutes les images en pixels HDR, dont la dynamique est une plage de valeurs dans un format supérieur à 8 bits par canal.

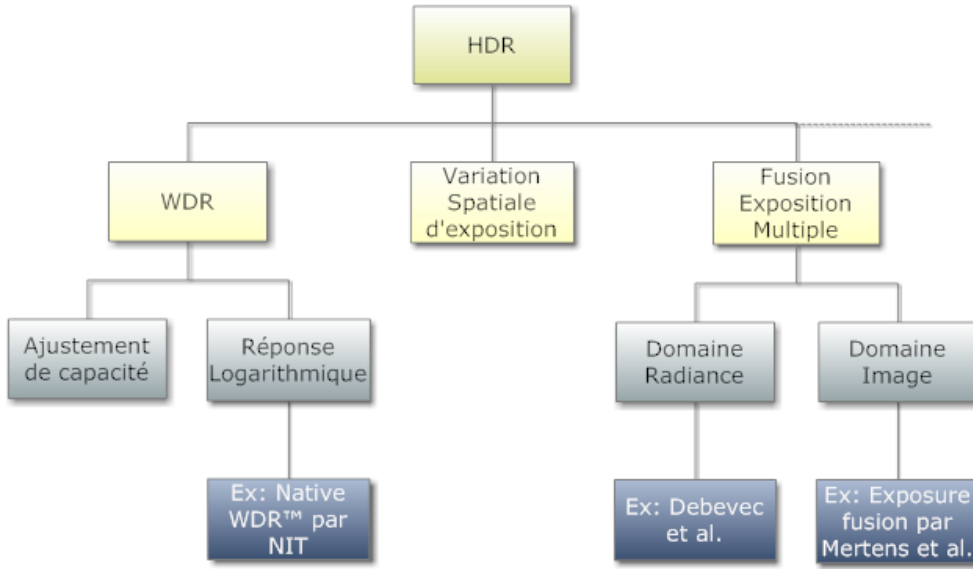


FIGURE 2.4 – Les différentes techniques HDR.

### 2.3.1.1/ RÉCUPÉRATION DE LA COURBE DE RÉPONSE

Dans une configuration idéale, créer une image HDR se fait simplement en normalisant les images à faible dynamique par leur temps d'exposition respectif, puis en additionnant les pixels des images normalisées correspondants. Cependant, du fait de la présence de circuits électroniques complexes, la plupart des caméras ont un traitement non linéaire vis-à-vis de la transformation de la lumière incidente. Cette transformation non linéaire est matérialisée par la courbe de réponse de la caméra, notée  $f$ . La caractéristique doit impérativement être connue avant la création d'une image HDR. Malheureusement, cette fonction est souvent issue d'une propriété intellectuelle, et est rarement fournie par le fabricant de capteur. La courbe de réponse de la caméra détermine donc la relation entre la lumière qui frappe le capteur et les valeurs de pixels en sortie de la caméra. Une illustration du système est présentée sur la Figure 2.5. En réalité, nous avons besoin de la courbe de réponse inverse du système  $f^{-1}$ . On fait l'hypothèse que la courbe de réponse est croissante et monotone sur l'intervalle considéré. De ce fait, si nous connaissons la fonction, nous connaissons aussi son inverse  $g = f^{-1}$ . Par exemple, la courbe de réponse de la caméra ayant servi à capturer les images dans la revue de Debevec [10], a été estimée par nos soins grâce à l'algorithme fourni dans la publication, elle est visible sur la Figure 2.6. Elle associe à chaque pixels  $Z_{ij}^p$  une valeur d'exposition telle que :

$$g(Z_{ij}^p) = \ln E_{ij} + \ln \Delta t_{ij}^p, \quad (2.6)$$

où  $i, j$  indexe la position du pixel, et  $\Delta t^p$  est le temps d'exposition de l'image  $p$ .

Pour la plupart des appareils, la fonction  $f$  est propre à chaque système (len-

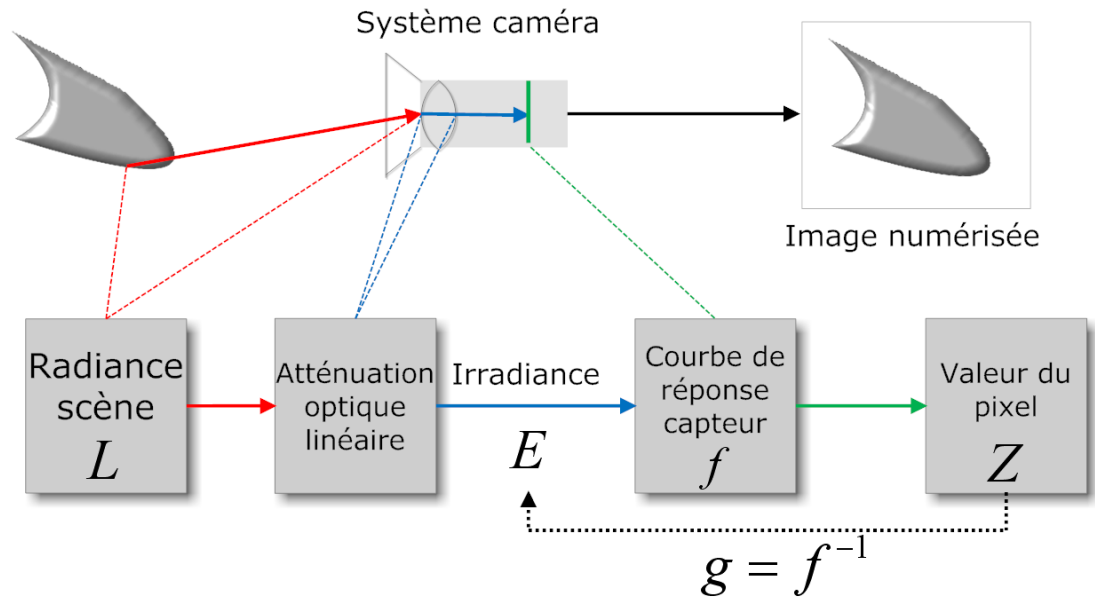
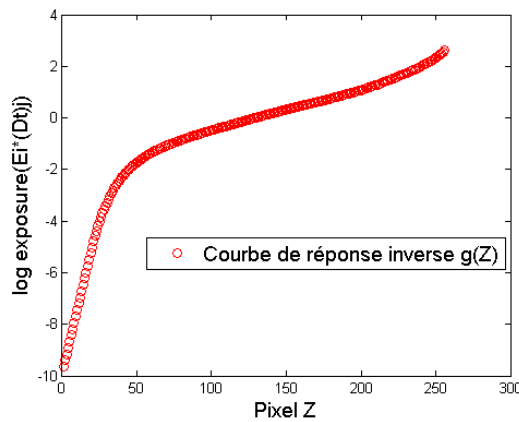


FIGURE 2.5 – Illustration du système de capture de radiance par une caméra.

FIGURE 2.6 – (a) Courbe de réponse inverse  $g(Z)$  donnant la relation  $g(Z) = \ln E \cdot \Delta t$ , pour le mémoriel de Debevec [10].

tille+ouverture+capteur), et doit être estimée du mieux possible pour reconstruire fidèlement les images HDR. Depuis 1995, plusieurs algorithmes ont vu le jour, permettant de retrouver cette fonction  $f$ .

Nous décrivons ici les 4 méthodes de la littérature les plus populaires dans le domaine HDR pour retrouver astucieusement cette fonction. Chacune propose également une fonction de pondération associée. Les fonctions de pondération seront résumées dans le Tableau 2.5.

1. Mann : *"Being 'undigital' with digital cameras : extending dynamic range by combining differently exposed pictures"* [40] Mann et al. ont été les premiers en 1995 à proposer une méthode permettant de regrouper les intensités lumineuses réelles



en format flottant dans une seule image HDR, à l'aide d'un set d'image numérisées. Le processus implante une reconstruction automatique de la courbe de réponse, à partir de multiple expositions. Une fois que l'on a associé la courbe de réponse à la caméra, on utilise la courbe de calibration pour la fusion. Néanmoins, cette méthode n'est pas reconnue comme donnant des résultats satisfaisants. La fonction de pondération utilisée par Mann et al. est la dérivée de la fonction de réponse du système  $f'^{-1}$ . Leur hypothèse s'appuie sur le fait que la validité des valeurs des pixels suit directement la sensibilité du système à la lumière.

2. Debevec : *"Recovering High Dynamic Range Radiance Maps from Photographs"* [10]

L'algorithme développé par Debevec et Malik [10] est basé sur le concept selon lequel la valeur du pixel  $Z$  est fonction de l'irradiance  $E$  perçue à travers l'objectif, et du temps d'exposition  $\Delta t$  auquel on soumet le système de capture. La fonction de réponse  $f$  de l'appareil est :

$$Z_{ij}^p = f(E_{ij}\Delta t_p). \quad (2.7)$$

L'idée clé est qu'une valeur d'irradiance  $E$  pour un temps d'exposition de  $2\Delta t$ , aura la même exposition qu'une valeur d'irradiance  $2E$  exposée pendant un temps  $\Delta t$ . Ce concept permet l'évaluation de la courbe de réponse de l'appareil, basée sur la résolution d'un ensemble d'équations linéaires par la décomposition en valeurs singulières (SVD). On suppose bien sûr que la courbe de réponse est continue. Lors de la récupération de chaque valeur de radiance à partir des expositions disponibles, ils utilisent une fonction de pondération simple de type "hat" (voir Figure 2.7), basée sur l'hypothèse que les pixels de milieu de gamme (proches de 128 pour un capteur 8 bits) sont les plus fiables et les mieux exposés.

3. Mitsunaga : *"Radiometric Self Calibration"* [12]

La méthode proposée par Mitsunaga et Nayar [12] calcule une fonction de réponse radiométrique estimée à l'aide d'un polynôme d'ordre élevé, sans connaissance préalable des temps d'exposition utilisés. Au cours du processus, les temps d'exposition sont estimés et sont affinés. Ceci en fait une approche intéressante, mais le processus est limité par le calcul d'un polynôme d'un certain ordre et des coefficients sélectionnés. Les auteurs affirment qu'il est possible de représenter pratiquement n'importe quelle courbe de réponse à l'aide d'un polynôme. De la même manière que la méthode de Debevec et Malik, on suppose que la courbe de réponse est continue. En ce qui concerne leur fonction de pondération, Mitsunaga et Nayar utilisent la théorie du signal de telle sorte que la valeur de pixel la plus élevée est moins influencée par le bruit de fond supposé constant.

4. Ward : *"High Dynamic Range Imaging : Acquisition, Display, and Image-Based Lighting"* [23]

Ward et al. réutilisent la courbe de pondération définie par Mitsunaga et al. en la multipliant par une fonction "hat" large, de telle manière à réduire l'effet imprévisible

que peuvent avoir les valeurs de pixels proches des extrêmes.

5. Robertson : *"Estimation-theoretic approach to dynamic range enhancement using multiple exposures"* [11]

Robertson et al. [11] proposent d'utiliser la dérivée de la fonction de la réponse de la caméra comme fonction de pondération, en mettant en avant le fait que la fiabilité des valeurs de pixel est en corrélation directe avec la sensibilité de l'appareil aux changements de lumière. De plus, l'hypothèse est faite que la sortie numérique se compose d'un bruit sous forme de Gaussienne. La méthode de Robertson et al. donne un facteur de pondération plus important aux pixels qui ont un temps d'exposition plus long, contrairement à la méthode de Debevec donnant un poids plus fort aux pixels de milieu de gamme. En effet, on décide de privilégier la fusion d'un pixel ayant un long temps d'exposition, plutôt qu'un temps d'exposition court. Cet algorithme est connu pour être un de ceux de l'état de l'art qui réduit le plus le bruit lors de construction d'une image HDR.

Le problème de la fonction de pondération est qu'elle doit être sélectionnée en fonction de l'application qui est visée.

Dans un article traitant du HDR en astronomie [41], l'approche est totalement différente à celle exposée ci-dessus, puisque l'on cherche plutôt l'information dans les parties extrêmes (intensités lumineuses infimes et très hautes luminosités) des radiances perçues du cosmos. On emploie donc une fonction de pondération radicalement différente si l'on s'intéresse plus particulièrement aux pixels proches de la zone de saturation. En outre, les valeurs de pixels proches du noir doivent ressortir, ce qui en fait un challenge intéressant qui donne à juger de l'acceptabilité du niveau de bruit du capteur destiné à l'astronomie.

### 2.3.1.2/ CONSTRUCTION DE LA CARTE DE RADIANCE

Une fois que la courbe de réponse est connue, elle est utilisée pour linéariser toutes les images prises par la caméra avec différents temps d'exposition  $\Delta t$ . Théoriquement, le nombre minimum d'images utilisées pour la reconstruction HDR est  $P = 2$ . Après cette étape, les pixels correspondants des images LDRs sont assemblés, pondérés et ajoutés afin de récupérer les valeurs réelles de radiances de la scène. Comme vu dans la partie précédente, la pondération permet de donner plus ou moins d'importance aux pixels pendant la somme des pixels, suivant si ils sont trop saturés, trop bruités, ne contenant que peu d'information, etc. La fonction de pondération est notée  $\omega$ . Voici la formule générale permettant d'estimer les valeurs d'irradiance :

$$\ln E_{ij} = \frac{\sum_{p=1}^P \omega(Z_{ij}^p) [g(Z_{ij}^p) - \ln \Delta t_p]}{\sum_{p=1}^P \omega(Z_{ij}^p)}. \quad (2.8)$$

Méthode	Hypothèse	Formule
1. Mann [40]	La fiabilité de la valeur du pixel est en corrélation avec la sensibilité de l'appareil aux changements de lumière	$\omega(Z) = \frac{1}{g'(Z)}$
2. Debevec [10]	Le pixel en milieu de plage est plus fiable	$\omega(Z) = \begin{cases} Z & \text{pour } Z \leq 127 \\ 255 - Z & \text{pour } Z > 127 \end{cases}$
3. Mitsunaga [12]	Le SNR augmente avec l'intensité du signal	$\omega(Z) = \frac{g(Z)}{g'(Z)}$
4. Ward [23]	Multiplication de la fonction de pondération de Mitsunaga par un filtre large de type "hat", pour exclure les pixels aux valeurs extrêmes	$\omega(Z) = \frac{g(Z)}{g'(Z)} \left(1 - \left(\frac{Z}{127.5} - 1\right)^{12}\right)$
4. Robertson [11]	La fiabilité des valeurs de pixel est en corrélation avec la sensibilité de l'appareil aux changements de lumière	$\omega(Z) = \frac{\Delta^2}{g'(Z)}$

TABLE 2.5 – Liste des fonctions de pondération existantes.

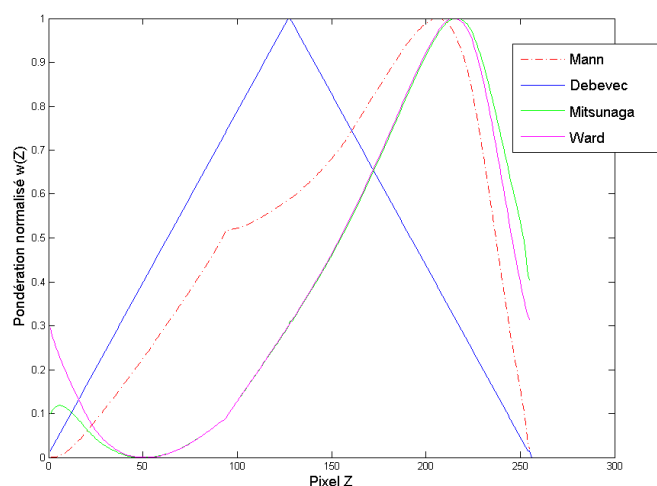


FIGURE 2.7 – Quatre fonctions de pondération normalisées [10, 12, 23, 40].

Les méthodes de pondérations ont fait l'objet d'études approfondies [42, 43]. Une des déductions faites annonce que l'algorithme de Debevec et Malik est le plus performant dans le domaine appliqué à la vidéo, et le moins sujet aux erreurs lors de la récupération de la courbe de réponse du système.

### 2.3.1.3/ REPRODUCTION DE TONS (TONE MAPPING)

Les algorithmes de reproduction de tons ou tone mapping (TMOs pour "Tone Mapping Operators"), servent à convertir une image HDR en une image affichable sur un écran standard. Il existe deux grandes catégories de TMOs : spatialement uniformes (TMOs globaux) et spatialement non-uniformes (TMOs locaux).

La manière la plus simple d'effectuer un tone mapping est d'appliquer un mapping linéaire qui compresse les luminances dans la gamme  $[0, 255]$  (pour un afficheur 8 bits). Par exemple, un mapping logarithmique compressera les valeurs de luminances dans la plage affichable désirée, après les avoir fait passer dans un domaine logarithmique.

Typiquement, un TMO global cherche le minimum et le maximum de luminance, ainsi que la luminance moyenne dans l'image HDR pour en réduire la plage de dynamique. Ce type d'opérateur est spatialement invariant, c'est à dire que l'on applique la même opération en tout points de l'image HDR. Des modèles faisant référence au domaine psychophysique, à l'adaptation visuelle, à la perception de contraste ou encore aux caractéristiques rétinienne forment la base théorique de ces méthodes. Les TMOs globaux sont simples, peu coûteux en ressources, et donnent en général des résultats satisfaisants. De plus, des méthodes globales plus évoluées existent telles que la compression logarithmique, l'ajustement d'histogramme, etc. Malgré tout, il peut subsister quelques effets indésirables, tels que le manque de contraste dans les zones fortement éclairées, ou des couleurs globalement délavées. Un exemple de tone mapping global de type logarithmique est présent sur la Figure 2.8. On peut remarquer plusieurs effets. Le contraste est assez faible dans la zone autour de l'ampoule, la zone à haute luminosité. Néanmoins, on peut remarquer que l'ensemble des informations pertinentes est représenté. L'ampoule en elle-même n'a pas de détail puisqu'elle est entièrement blanche ; seulement ceci n'est pas dû au tone mapping, mais plutôt aux images que l'on a utilisées pour construire la carte de radiance. En effet, si l'exposition basse n'est pas assez faible, le capteur reste tout de même saturé dans les blancs et ceci se voit au final sur l'image HDR après tone mapping. En outre, cette image comporte de bons contrastes dans les zones de basses luminosités. Nous verrons par la suite que ces contrastes peuvent différer selon l'espace choisi entre les différents temps d'exposition.

Un tone mapping local, contrairement au TMO global, prend en considération les pixels voisins du pixel HDR à "tone mapper", et leur donne un impact direct sur l'évaluation de la valeur du pixel après tone mapping. La méthode est inspirée du système visuel humain, et cherche notamment à reproduire le comportement des récepteurs visuels de type cônes et bâtonnets. D'autres algorithmes complexes, s'appuient sur des théories telles que :

- Le Retinex [44]. La théorie prédit l'adaptation humaine à l'intensité lumineuse. Les auteurs mettent en œuvre une variation Retinex définissant un entourage pixelique de type gaussien, pour arriver à réaliser à la fois une compression de dynamique, la "color



**FIGURE 2.8** – Image résultante d'un tone mapping global de type logarithmique.

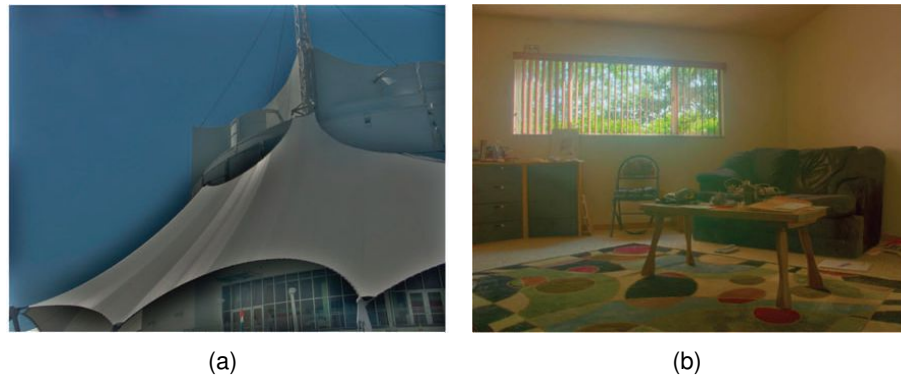
constancy" et un bon rendu des luminosités.

- Le mode d'observation multi-échelles de luminance [45]. Le modèle fournit un cadre cohérent pour la compréhension de l'effet de l'adaptation de la vision spatiale. Il prend en compte le seuil de visibilité, l'acuité visuelle et la discrimination des couleurs et la sensibilité aux contrastes se produisant pendant les changements de niveaux d'éclairage dans la scène.
- La photographie [14] avec la méthode de "Dodging and Burning" initialement utilisée en photographie argentique. Le "dodging" augmente l'exposition pour les zones de l'image où le photographe veut plus de lumière. Le "burning" baisse l'exposition dans les zones qui doivent être plus sombre.
- Le modèle d'apparence des couleurs iCAM [46]. Plus complexe que les derniers modèles, puisqu'il n'inclut pas uniquement une compression de dynamique, mais décrit plutôt un ensemble de modules qui constituent un modèle complet, dont le filtrage spatial des artefacts, l'évaluation de qualité, le gamut, etc.

Plusieurs anomalies peuvent apparaître à l'usage de ces algorithmes. D'une part, ils sont assez complexes à mettre en œuvre et nécessitent beaucoup de ressources matérielles, si on s'en réfère aux architectures existantes qui les implémentent. D'autre part, ils peuvent engendrer des effets visuels indésirables sur le rendu final de l'image après tone mapping, de type halos ou perte des contours ; dans le cas où l'on n'appliquerait pas de filtres et de corrections supplémentaires, elles aussi consommatrices de ressources. Une illustration qui montre les effets indésirables qui peuvent survenir dans le cas de l'application d'un tone mapping local est présentée sur la Figure 2.9.

Dans le cas d'une vidéo HDR, plusieurs algorithmes semblent être implémentables en temps réel grâce à des temps de calcul faibles, qu'ils soient globaux ou locaux. Dans le Tableau 2.6, nous présentons une liste non exhaustive des méthodes de tone mapping dont l'efficacité et la simplicité des calculs sont démontrés dans les publications relatives au HDR.

Le TMO local de Reinhard et al. [14] de reproduction photographique des tons a déjà fait l'objet de travaux hardware temps réel [49]. Malheureusement, l'algorithme a besoin



**FIGURE 2.9** – Illustration des effets indésirables de type halos, pouvant apparaître lors de l'utilisation de TMO locaux de Chiu et al. [47] (a) et Yee et al. [48] (b). Les images sont tirées du livre de Reinhard [23].

d'une image HDR entière en entrée pour fonctionner, ainsi que de beaucoup de mémoire. Nous aurons l'occasion de pouvoir étudier cet algorithme dans la description des systèmes de vision dans le Chapitre 3, dédié à l'état de l'art.

D'après la publication de 2010, l'algorithme de Duan et al. permet un calcul massivement parallèle, pour obtenir au final une image 5 à 6 fois plus rapidement comparé aux techniques de Reinhard et al. [50] et Fattal et al [17]. Une autre méthode [51] implémentée sur GPU ("Graphics Processing Unit") permet l'accélération des TMOs globales grâce à un échantillonnage de l'image HDR d'entrée.

Afin de faciliter la recherche dans le domaine, Mantiuk et al. [52] ont créé en 2007, un logiciel open source nommé *pfstools*, qui regroupe les sources de plusieurs algorithmes de création HDR et de tone mapping [52]. Tous les algorithmes sont disponibles en C++. Ce logiciel est utilisé pour l'encodage de vidéo HDR, les applications médicales, la manipulation de textures, etc. Il est possible de communiquer avec différents langages tels que le C++, Java, Matlab et GNU Octave. L'outil intègre aussi les deux algorithmes de génération HDR, Robertson et al. [11] et Mitsunaga et al. [12].

Dans le cadre de cette thèse, nous développerons nous-mêmes un outils nommé *HDR-Lab*, décrit dans la Section 4.5.1, capable d'évaluer des algorithmes HDR, notamment par l'intégration d'outils de mesure de qualité, d'analyse statistiques ou de comparaison sous Matlab.

#### 2.3.1.4/ L'ÉVALUATION DES OPÉRATEURS DE REPRODUCTION DE TONS

Représenter le plus fidèlement possible la réalité est un aspect intéressant à prendre en compte pour réaliser notre projet. Plusieurs publications ont été faites concernant l'évaluation des méthodes de tone mapping. Une étude [56] montre que la méthode Reinhard et al. [14] de tone mapping photographique donne les meilleurs résultats en termes d'as-

Quelques opérateurs de reproduction de tons.			
<i>Méthodes TMO</i>	<i>Année</i>	<i>Titre</i>	<i>Type</i>
Pattanaik et al. [45]	2000	Time-Dependent Visual Adaptation for Realistic Image Display	Global
Reinhard et al. [14]	2002	Photographic Tone Reproduction for Digital Images	Global & Local
Durand et al. [15]	2002	Fast Bilateral Filtering for the Display of High-Dynamic-Range Images	Local
Ashikhmin et al. [53]	2002	A Tone Mapping Algorithm for High Contrast Images	Local
Fattal et al. [17]	2002	Gradient Domain High Dynamic Range Compression	Local
Drago et al. [13]	2003	Adaptive Logarithmic Mapping for Displaying High Contrast Scenes	Global
Reinhard et al. [54]	2003	Dynamic range reduction inspired by photoreceptor physiology	Global
Mantiuk et al. [55]	2006	Perceptual Framework for Contrast Processing of High Dynamic Range Images	Local
Duan et al. [50]	2010	Tone-mapping high dynamic range images by novel histogram adjustment	Global & Local

TABLE 2.6 – Quelques opérateurs de reproduction de tons.

pect naturel de l'image. Une autre étude de Drago et al. [57] arrive à la même conclusion. De plus, l'étude de Cadik et al. [58] montre de très bons résultats pour l'opérateur de Reinhard [14]. Il existe un aspect étonnant dans cet article : la partie globale des méthodes de tone mapping est essentielle pour l'obtention de bons résultats visuels pour les scènes du monde réel. Le graphique de la Figure 2.10 met en évidence les résultats de cette étude.

Ledda et al. [59] ont fait une étude psychologique afin d'évaluer 6 TMOs [13–15, 60–62] dans des conditions réelles de visualisation. Ils utilisent pour cela des images HDR, affichées à la fois sur écran HDR comme références, et sur deux écrans simples après tone mapping. L'utilisateur doit choisir entre les deux images affichées, suivant plusieurs critères (similitude globale, détails clairs et détails sombres). Ils essayent de définir ainsi le "meilleur" algorithme du point de vue du spectateur, et définissent un procédé expérimental pour l'évaluation des TMOs. Un résultat intéressant qui ressort lorsque le spectateur visionne une image couleur dans son ensemble, est que l'apparence des couleurs semble dominer dans son choix, même si il existe un manque de contraste certain dans l'image. Une autre expérience est faite sur la perception des détails, indépendamment sur les régions claires et sombres des images. Les auteurs en concluent que l'algorithme proposé par Johnson et al. [60] basé sur iCAM [46] ("Color Appearance Model") met bien en valeur les détails dans les zones fortement éclairées. De même, les expériences

montrent que l'algorithme de Reinhard et al. [14] reproduit le mieux les endroits sombres de la scène HDR.

D'après l'évaluation perceptive de sept TMOs par Yoshida et al. [63], un algorithme doit avant tout être choisi en fonction des besoins de l'application visée. La qualité des TMOs peut varier selon le type d'éclairage, l'intensité d'éclairage de la scène, les paramètres utilisateur, etc. D'autres aspects tels que le degré de détails, le marquage des contours et l'aspect naturel du rendu sont également à prendre également en compte.

Le but, dans notre projet, est d'obtenir le meilleur rendu possible, tout en prenant compte les dispositions matérielles. Puisqu'il n'existe pas de méthode standardisée de tone mapping, un logiciel nommé *HDR-Lab* a été créé, afin d'évaluer les algorithmes potentiellement utilisables pour le temps réel. La description de cet outil et le choix des algorithmes se feront plus tard dans les Sections 4.5.1 et 4.5.2.

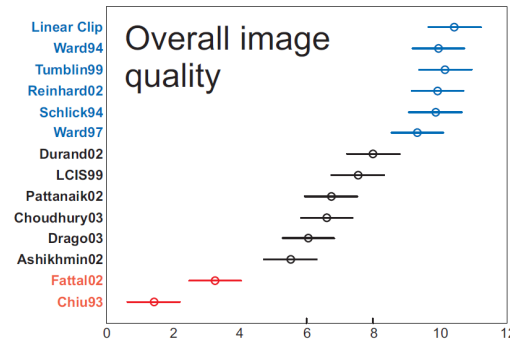


FIGURE 2.10 – Résultats de la qualité d'images produits par l'étude de Cadik et al. [58]

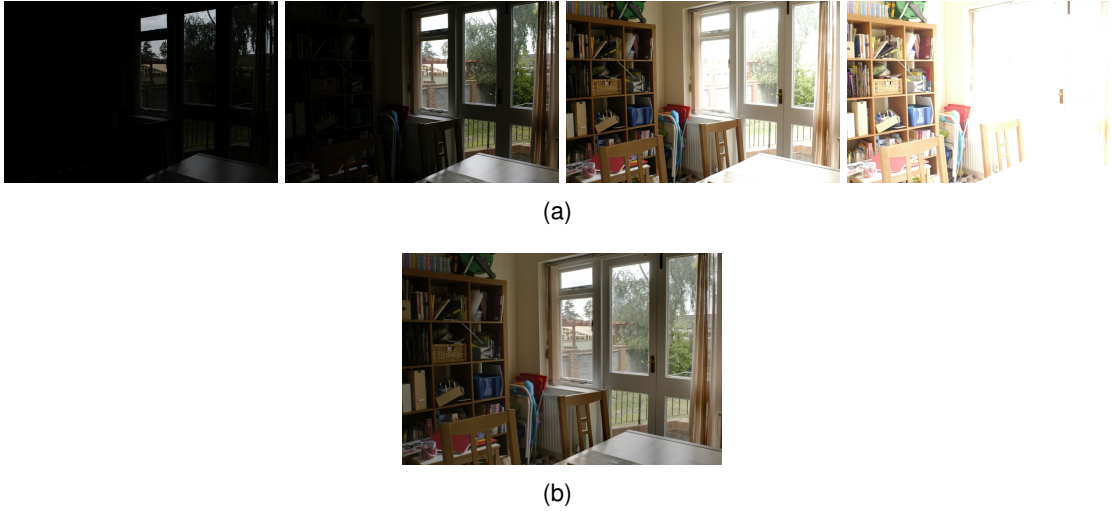
### 2.3.2/ FUSION D'EXPOSITIONS DANS LE DOMAINE IMAGE

L'*exposure fusion* est une méthode connexe à la méthode usuelle de création HDR. Il est possible d'assembler des images directement dans le domaine à 8 bits par canal, et d'effectuer le rendu sans préalablement passer par les étapes de création d'image HDR et de tone mapping. La fusion d'exposition peut sélectionner les meilleures parties exposées des différentes images LDRs en entrée, et les fusionner ensemble pour former une image bien exposée. Toutefois, la technique de capture des images LDRs est identique à celle utilisée par la méthode HDR dans le domaine des radiances. Plusieurs démarches de fusion d'exposition vont être détaillées dans la suite de cette section.

En 2005, Goshtasby et al. [65] ont proposé une nouvelle méthode pour fusionner des images d'expositions multiples. Chaque image est partitionnée en blocs uniformes. Pour chaque petit bloc dans chaque image, ils identifient le meilleur bloc qui a la meilleure valeur d'entropie. L'entropie est calculée de la manière suivante :

$$E_n = \sum_{i=0}^{255} -p_i \log(p_i), \quad (2.9)$$





**FIGURE 2.11** – Un exemple de la technique de fusion d'exposition proposée par Mertens et al. [64]. Un set d'exposition multiples (a), et un résultat HDR-like sans passer par la création, à proprement parlé, d'une image HDR et par le tone mapping (b). Les images sont de Mertens et al. [64]

où  $p_i$  est la probabilité pour qu'un pixel dans l'image est une intensité  $i$ . Ensuite, les blocs sélectionnés sont assemblés par l'intermédiaire d'une fonction monotone. Ils déterminent aussi la taille de bloc, par l'intermédiaire d'une fonction récursive, afin d'optimiser la détection d'information dans les images. Malheureusement, cette méthode demande beaucoup de ressources. Le temps de calcul est relativement long, et varie selon la taille des images en entrée. Cette méthode est donc assez difficile à intégrer sur un système embarqué, et est peu appropriée à une implémentation en temps réel. Cependant, ce travail a pour avantage de bien savoir manipuler les contours des objets dans la génération du résultat HDR-like, chose loin d'être évidente lorsque l'on fusionne des images dans le domaine image.

Une nouvelle technique a ensuite été proposée par Mertens et al. [66], obtenant de bons résultats lors de la fusion d'images. Contrairement à la technique de Goshtasby [65], la fusion est réalisée au niveau pixelique, et non par blocs. Le processus de fusion attribue un poids pour tout les pixels de l'image, relatif à des facteurs de qualité tels que l'entropie, la saturation, le contraste, ou encore la bonne exposition des pixels. Une cartographie de poids est entièrement calculée pour chaque image LDR. Ces poids influent par la suite sur le degré d'inclusion ou d'exclusion du pixel de telle ou telle image dans le résultat final HDR-like. D'après l'article [66], on obtient des images très naturelles tout en diminuant les temps de calcul. La fonction de pondération peut être écrite ainsi :

$$W_{k,ij} = (C_{k,ij})^{\omega_C} \times (S_{k,ij})^{\omega_S} \times (E_{k,ij})^{\omega_E}, \quad (2.10)$$

où  $C$ ,  $S$  et  $E$  sont respectivement le contraste, la saturation et l'état d'exposition du pixel correspondant.  $i$  et  $j$  indexent la position du pixel dans l'image, l'image étant elle indexée

par  $k$ . Durant le processus de fusion,  $W_{ij,k}$  est utilisé pour sélectionner le pixel approprié, conformément aux métriques de qualité. Ensuite, une normalisation des valeurs est effectuée sur les  $N$  cartes de pondération, en les ajoutant les unes aux autres et ce, à chaque position de pixel :

$$\hat{W}_{k,ij} = \left[ \sum_{k'=1}^N W_{k',ij} \right]^{-1} W_{k,ij}. \quad (2.11)$$

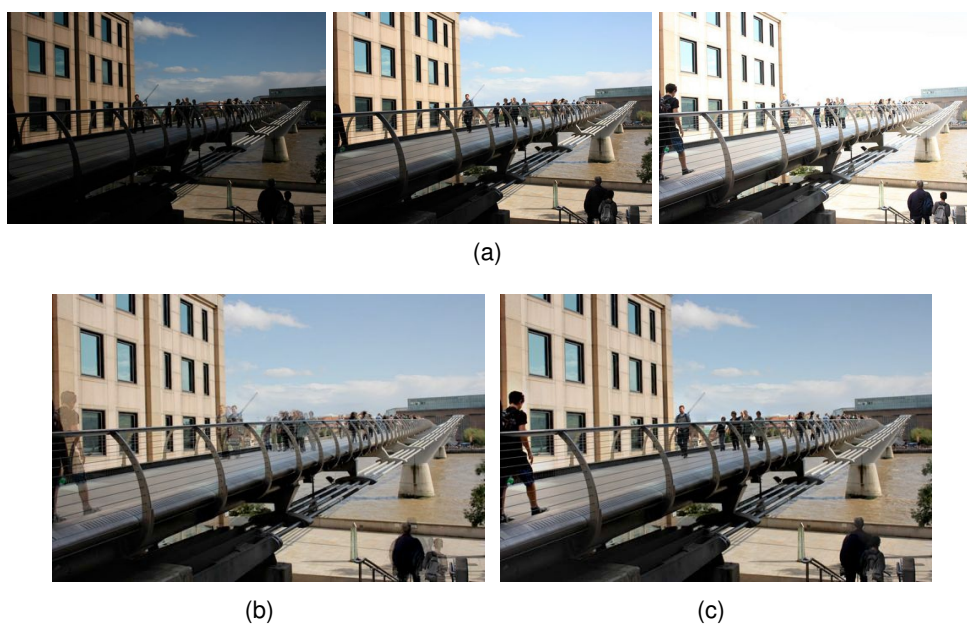
La fusion des images LDRs se fait ensuite en utilisant une décomposition de Laplace et une pyramide Gaussienne des cartes de pondération comme ceci :

$$L\{R\}_{ij}^l = \sum_{k=1}^N G\{W\}_{k,ij}^l L\{I\}_{k,ij}^l, \quad (2.12)$$

où  $G\{W\}_{k,ij}^l$  est la pyramide Gaussienne des cartes de pondération, et  $L\{I\}_{k,ij}^l$  est la décomposition Laplacienne en chaque niveau  $l$ . Ensuite, la pyramide fusionnée  $L\{R\}^l$  pour chaque résolution est réduite afin d'obtenir le résultat final  $R$ .

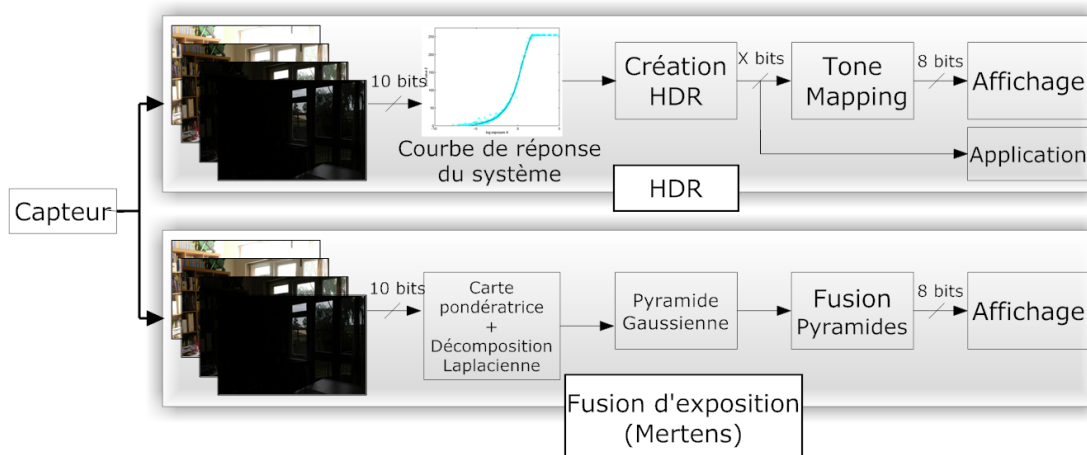
En 2010, une extension [67] de l'algorithme de Mertens [64] a été faite afin d'optimiser la correction de mouvement entre les différentes images acquises. La méthode détecte les pixels en mouvement entre chaque exposition, et permet de supprimer les effets indésirables lors de la génération de l'image. La méthode fonctionne bien pour des mouvements situés en fond de scène, ou ayant de petites dynamiques de mouvement. La détection de dynamique se fait par de simples opérations binaires, ce qui la rend propice à une implémentation hardware. Cependant, l'aspect temps réel demande encore à être investigué, notamment pour trouver une solution d'inclusion de l'algorithme dans un contexte de calcul massivement parallèle. Un exemple de résultat de fusion avec cette méthode est visible sur la Figure 2.12.

Pour conclure, la fusion d'exposition est une technique HDR-like récemment apparue, qui permet de travailler uniquement dans le domaine de l'image sans avoir à générer une carte de radiance et à compresser la dynamique des luminances de l'image HDR. La représentation des pixels est donc assez simple tout au long du processus. Toutefois, des calculs mettant en œuvre des opérateurs et des fonctions complexes sont nécessaires. En outre, l'utilisation inévitable d'opérateurs flottants et/ou d'opérateurs à virgule fixe fait perdre de l'intérêt à la fusion d'exposition par rapport à la technique originale HDR. Les limites se font aussi bien sentir au niveau de la diversité des domaines d'applications restreints. Par exemple, l'interopérabilité avec les futurs écrans HDRs ne pourra malheureusement pas se faire, du fait du manque de dynamique dans la représentation des données. De plus, le résultat généré n'est pas radiométriquement correct. En effet, la fusion d'exposition ne permet pas d'estimer directement les valeurs de radiances réelles d'une scène, ce qui est l'intérêt propre de la technique HDR. Ceci peut limiter l'utilisation de cette technique pour les applications d'infographie utilisant la technique "Image-based lighting" par exemple. Néanmoins, il est tout de même intéressant de suivre l'évolution



**FIGURE 2.12** – Illustration de la technique de fusion d'exposition pour des scènes dynamiques. (a) montre une séquence d'acquisition, (b) le résultat de fusion avec la technique originale de Mertens [64] et (c) le résultat de la technique de Pece et al. [67]

de la fusion d'exposition dans le domaine de la vidéo et du temps réel, car le nombre de publications scientifiques en rapport avec cette technique augmente ces derniers temps. Les différences fondamentales entre les pipelines HDR et de fusion d'exposition sont résumées sur la Figure 2.13.



**FIGURE 2.13** – Différence au niveau des pipelines des méthodes de fusion d'exposition et HDR.

## 2.4/ PROBLÈMES LIÉS AUX VIDÉOS HDR

### 2.4.1/ ALIGNEMENT DES IMAGES

Lorsque l'on capture des images successives avec des temps d'exposition différents, le mouvement de la caméra pendant l'acquisition peut entraîner des effets indésirables sur le résultat HDR. Il paraît alors judicieux d'aligner les images LDRs avant d'effectuer la génération HDR. Ceci en va de même pour la méthode de fusion d'exposition.

La méthode bien connue pour aligner les images est celle de Ward [68]. Une pyramide MTB ("Median Threshold Bitmap") est générée à partir des images en entrée. La bitmap attribue la valeur '0' aux pixels d'entrée qui sont inférieurs ou égaux à la valeur médiane, et '1' lorsque les valeurs de pixels sont plus grandes. Le décalage global pour l'alignement est calculé à partir de la paire MTB de résolution la plus basse, en testant un décalage dans la gamme  $\{-1, 0, 1\}$ , dans les directions horizontale et verticale. Au niveau de résolution suivant, ce décalage est multiplié par 2, et le résultat est testé avec sa paire décalée dans le domaine  $\{-1, 0, 1\}$ , et dans les directions mentionnées. Ce processus continue jusqu'au niveau le plus élevé, conduisant à l'alignement final des images.



**FIGURE 2.14** – Une pyramide MTB (Median Threshold Bitmap) (b) est utilisée pour l'alignement des images d'entrée (a) proposé par Ward [68].

Dans le cas d'un système de vision, il n'est pas aisé d'appliquer un tel algorithme. Si on veut minimiser l'effet dû à l'alignement des images LDRs, il est important de bien choisir le capteur mis en jeu. Un capteur trop peu sensible aura un temps d'acquisition trop

long pour une même exposition, et donc délivrera une sortie peu cadencée en termes de nombre d'images par seconde. Un système pouvant fonctionner à plus de 50 images/s permet de limiter les défauts d'alignement pour des mouvements de caméra raisonnablement faibles.

#### 2.4.2/ DÉTECTION ET ÉLIMINATION DE "GHOST"

La multiple exposition présente certains désavantages pouvant survenir si un mouvement est présent pendant la capture des images à fusionner. Contrairement au problème d'alignement des images énoncé précédemment, les ghosts ne sont pas des problèmes à traiter globalement dans l'image, mais bel et bien localement, ce qui rend la tâche encore plus difficile. Et elle est d'autant plus difficile si les deux problèmes interviennent en même temps et sont corrélés.

La correspondance des pixels faite lors de la génération HDR peut entraîner l'apparition d'artefacts nommés "ghosts" sur l'image résultante. Cela se remarque plus particulièrement dans une scène extérieure où le vent ferait bouger les feuilles des arbres par exemple. Une des techniques simples d'élimination de ghost consiste en plusieurs points :

1. Partitionner les multiples images en blocs
2. Calculer la variance pondérée pour chaque bloc
3. Sélectionner les blocs pour lesquels la variance pondérée dépasse un certain seuil
4. Remplacer les blocs par les pixels d'un des blocs d'une seule exposition
5. Générer l'image HDR.

Seulement, cette méthode n'est pas des plus efficaces, car elle ne permet pas d'éliminer proprement les ghosts lorsque l'objet en mouvement a pour arrière plan une couleur similaire à la couleur qui le compose. D'autres méthodes ont par la suite été proposées, telles que la détection basée sur l'entropie [69], sur l'ordonnancement des pixels [70], les histogrammes, la détection à seuillage multi-niveaux [71], la détection par "bitmap" [67], etc. Certaines ne traitent que des mouvements locaux d'une scène, en occultant la suppression de ghosts due au mouvement de la caméra.

Comme le disent Srikantha et Sidibé [72], le nombre d'images LDRs en entrée prend une place importante lors de la sélection de l'algorithme de détection et de suppression de ghosts. En effet, certains algorithmes donnent de bons résultats pour un petit nombre d'images, alors que d'autres, basés sur la statistique, requièrent un plus grand nombre de données pour fonctionner correctement. De plus, l'utilisation de tel ou tel algorithme sur un grand nombre d'images augmente plus ou moins le temps de calcul respectif.

À notre connaissance, il n'existe pas de méthode temps réel pour la détection et l'élimination de ghosts dédiée à la multiple exposition.



**FIGURE 2.15** – Exemple d'image HDR issue du livre de Bloch [73]. On peut y déceler le problème de ghost énoncé dans la Section 2.4.2.

## 2.5/ CONCLUSION

Nous avons étudié, dans ce chapitre, l'aspect algorithmique de la génération et du rendu HDR dans le cas d'une multiple exposition. Avant d'envisager de construire une carte de radiance d'une scène représentative des vraies valeurs physiques de lumière, il a fallu définir les principaux aspects théoriques relatifs à la répartition dynamique de la lumière dans un milieu. Après avoir constaté la limite technique omniprésente apparaissant sur les capteurs d'images contemporains : le *manque de dynamique*, nous avons ensuite répertorié les deux façons de procéder afin d'augmenter artificiellement la dynamique des capteurs. Lorsque l'on opère dans le domaine 8-bit, on parle de fusion d'exposition. Cette technique n'est pas vraiment une technique HDR à proprement parlé ; elle sélectionne plutôt les pixels au sein des différentes images LDRs, en les fusionnant après une pondération effectuée suivant des critères présélectionnés. Cette technique limite les types d'application envisageables à la suite d'un traitement HDR. L'autre technique est celle que l'on appelle communément HDR, qui cherche à représenter les valeurs de radiance réelles tirées d'une scène. On lui associera la technique de tone mapping, servant à visualiser l'information pertinente sur un écran de faible dynamique. C'est cet assemblage de techniques qui va plus particulièrement nous intéresser dans les trois chapitres suivants. De ces méthodes découlera un système caméra, capable de fusionner des pixels en temps réel à une grande cadence.

Le prochain chapitre sera constitué de deux parties bien distinctes :

1. L'état de l'art des systèmes globaux/partiels et temps réel/non temps réel de vision



## HDRs

2. La sélection matérielle pour une caméra dédiée à la vision HDR, pouvant intégrer un pipeline d'algorithmes complet et fonctionnel que l'on définira.

# ÉTAT DE L'ART : VERS UN SYSTÈME MATÉRIEL TEMPS RÉEL DÉDIÉ HDR

## 3.1/ INTRODUCTION

Le problème de capture de dynamique complète d'une scène réelle, allant de la génération d'images HDR à la technique de réduction de dynamique de l'image, a attiré l'attention de nombreux chercheurs. Cependant, ces travaux ont trop peu souvent fait l'objet d'implémentation des algorithmes en hardware, nécessitant la résolution de problèmes inhérents au système matériel. La plupart des travaux software ne sont généralement pas adaptés à l'implémentation sur plates-formes matérielles telles que les FPGA ("Field Programmable Gate Array") ou les ASICs ("Application-Specific Integrated Circuit"). Dans ce chapitre, nous étudierons les solutions existantes relatives à la mise en place d'algorithmes dédiés à la vision HDR sur plates-formes de toutes sortes. Nous arriverons à une conclusion globale sur l'ensemble des systèmes existants, qu'ils soient temps réel ou non, complets ou non. L'état de l'art aura pour but de servir de base à la réalisation d'un prototype hardware de vision HDR. Nous nous attarderons plus particulièrement sur les techniques d'extension de dynamique des capteurs (Section 3.2), sur les systèmes traitants essentiellement des problèmes de captures (Section 3.4), ou bien des systèmes traitement uniquement de génération HDR (Section 3.5) ou encore de rendu du contenu à grande dynamique (Section 3.6). Nous énoncerons, dans la Section 3.7, les systèmes hardware existants, temps réel ou non. S'appuyant sur cet état de l'art, nous finirons par décrire notre choix matériel en détails en fin de chapitre, au travers d'une nouvelle caméra HDR, intitulée *HDR-ARtiSt* dans la Section 3.8.

## 3.2/ TECHNIQUES POUR ÉTENDRE LA DYNAMIQUE D'UN CAPTEUR

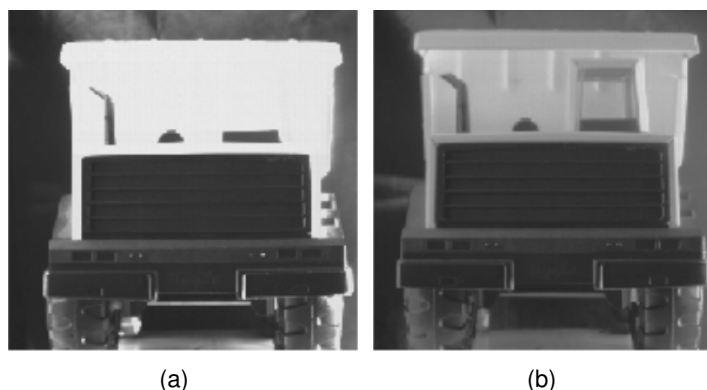
Les capteurs CMOS conventionnels ont généralement une dynamique s'étalant de 55 à 65dB. Il existe des sources de bruits associés aux capteurs d'images. L'une des plus importantes est le "dark current", le signal généré lorsqu'il n'y a pas de lumière. Les autres, telles que le FPN, les pixels chauds ou froids, le bruit d'amplification, sont en-



core d'autres sources de bruit limitant la dynamique des capteurs. L'augmentation de la dynamique peut se faire soit en faisant diminuer le "dark current", soit en augmentant le niveau de saturation du pixel. Néanmoins, d'autres moyens ont petit à petit vu le jour. Nous détaillerons ici brièvement les principaux procédés ingénieux, s'appuyant sur des concepts matériels et de conception, servant à élargir cette dynamique des capteurs.

### 1. Capteur à ajustement de capacité

En 1998, Dekker et al. [74] ont développé un nouveau type de capteur APS dits "Wide Dynamic Range" (WDR), d'une résolution de  $256 \times 256$  pixels. Le système est capable d'augmenter la dynamique du capteur standard d'environ  $20\times$ , soit une augmentation d'environ  $26\text{dB}$ , ou une dynamique finale proche de  $96\text{dB}$ . Une illustration du résultat final est présent sur la Figure 3.1. Le circuit numérique, composé de convertisseurs ADCs implantés par colonnes et en parallèle, fournit une cadence d'image entre 30 et 390 FPS. Il y a 128 circuits de sortie, chacun d'entre eux composé de deux sources de courant de polarisation. La grande dynamique est réalisable grâce à un ajustement de reset plusieurs fois durant le temps d'une intégration. Le courant de charge de la fonction de transfert (charge  $Q$  en fonction du courant  $i$ ) est comprimé, ce qui revient à élever le courant maximum  $i_{max}$  saturant le système. La mise en œuvre est assez simple mais malheureusement le système provoque du FPN et du bruit supplémentaire.



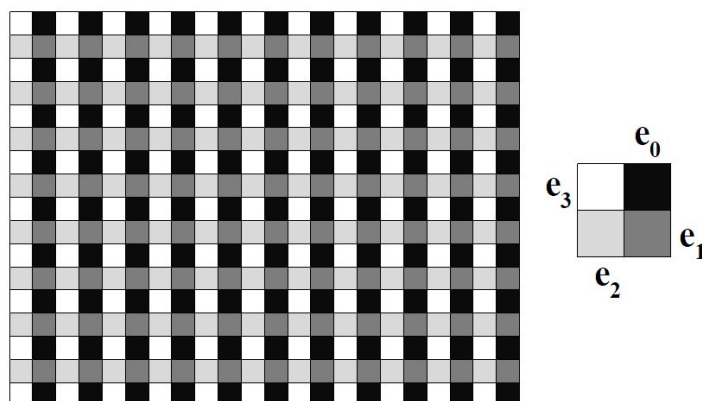
**FIGURE 3.1** – Technique d'extension de dynamique initiée par Dekker et al. [74]. L'image (a) est issue d'une capture linéaire (70 dB), tandis que (b) est une image résultante de la méthode de compression proposée (96 dB).

### 2. Variation spatiale d'exposition

Une autre technique d'amélioration de la plage dynamique des capteurs conventionnels est la variation spatiale d'exposition [3]. On sacrifie la résolution spatiale en interposant une matrice de filtre par dessus le capteur (voir Figure 3.2). Les valeurs de pixel de  $e1$ ,  $e2$ ,  $e3$  et  $e4$  correspondraient en fait aux 4 différentes images LDRs que l'on utiliserait pour faire la fusion HDR. À la manière d'un filtre de Bayer, la capture des différents niveaux d'illumination se fait en une simple intégration.

D'une part, les filtres sombres permettent de capturer de façon efficace les valeurs de radiances correspondantes aux zones hautement éclairées. De la même manière, les deux autres filtres moins opaques, médians et clairs, permettent d'isoler les luminosités des zones à éclairage moyen et bas. L'image de haute dynamique est alors générée en externe avec des techniques d'interpolation ou bien des filtres passe-bas. Le problème ici est qu'une grande quantité de lumière est réfléchiée sur la matrice, ce qui tend à augmenter le bruit et diminuer la sensibilité du capteur d'origine. De plus, on aura besoin d'utiliser un capteur de résolution bien plus grande si l'on veut garder une cohérence dans le rendu et la définition de l'image.

Récemment, un brevet [75] a été déposé par Aptina, présentant une méthode similaire, mais où chaque image capturée est construite de lignes entrelacées de deux expositions différentes. Par interpolation et correspondance entre les images, ils appliquent des traitements supprimant les effets de ghost dans l'image HDR résultante, lors de la fusion des images entrelacées. Rien n'est dit quant à la technique de capture des images entrelacées.



**FIGURE 3.2** – La technique de variation spatiale d'exposition. Les pixels peuvent être simultanément échantillonnés afin de capturer toute la gamme de luminance de la scène par petits morceaux.

### 3. "Time-to-saturation"

D'autres capteurs sont fabriqués suivant la technique "time to saturation", comme présentée dans l'article de Guo et al. [76]. L'idée est de mesurer le temps d'intégration nécessaire pour saturer chaque pixel de la matrice du capteur. Si nous considérons une quantité  $Q_{sat}$  (FWC pour " Full Well Capacity") de charges négatives accumulées jusqu'à la saturation du pixel,  $q$  la valeur d'une charge électrique,  $t_{sat}$  le temps d'exposition pour arriver à saturation, le photocourant  $i_{ph}$  est donné par la formule :

$$i_{ph} = \frac{qQ_{sat}}{t_{sat}}. \quad (3.1)$$

$Q_{sat}$  étant supposée fixe pour un type de capteur, on mesure donc le temps  $t_{sat}$  nécessaire à la saturation de la capacité. Pour ce type de capteur, on utilise la vitesse

d'obturation minimale pour saturer tous les photosites. Au moment où un photosite atteint sa pleine capacité, il enregistre une information qui indique combien de temps il a fallu pour atteindre cette capacité. Il en résulte un tableau 2D contenant les temps de saturation pour chaque pixel.

#### 4. Capteur logarithmique

Kavadias et al. [2] ont initialement présenté un capteur à réponse logarithmique. Le photocourant est directement converti en tension pour la lecture en sortie. Le gain en dynamique est réalisé lors de la conversion en tension de sortie  $v_o$ , par une compression logarithmique du courant telle que :

$$v_o = k \times \ln \frac{i}{i_0}, \quad (3.2)$$

où  $i$  est le courant qui passe dans la photodiode,  $i_0$  le courant d'obscurité et  $k$  une constante. On utilise en fait la forme exponentielle qu'a la source de courant dans un MOSFET. Plus précisément, la décroissance du courant après le pic de recouvrement suit une loi exponentielle. Cette technique n'est néanmoins pas dénuée d'inconvénients. En effet, le FPN est significativement augmenté. Le comportement non linéaire ne permet pas l'utilisation de CDS ("Correlated Double Sampling") pour corriger le FPN.

Yang Ni et al. [77] ont développé en 2011 un nouveau type de capteur logarithmique qui utilise le mode photovoltaïque de la photodiode. Le capteur CMOS a une résolution de  $768 \times 576$  pixels et une gamme dynamique mesurée de  $120dB$ . Leur nouvelle conception de pixel de taille  $10\mu m \times 10\mu m$  permet de réduire grandement le FPN ("Fixed Pattern Noise") comparé aux autres capteurs logarithmiques, et d'augmenter la sensibilité. Le capteur ne comporte par ailleurs pas de contrôle d'exposition. Des scènes de haute gamme dynamique sont visibles sur la Figure 3.3.



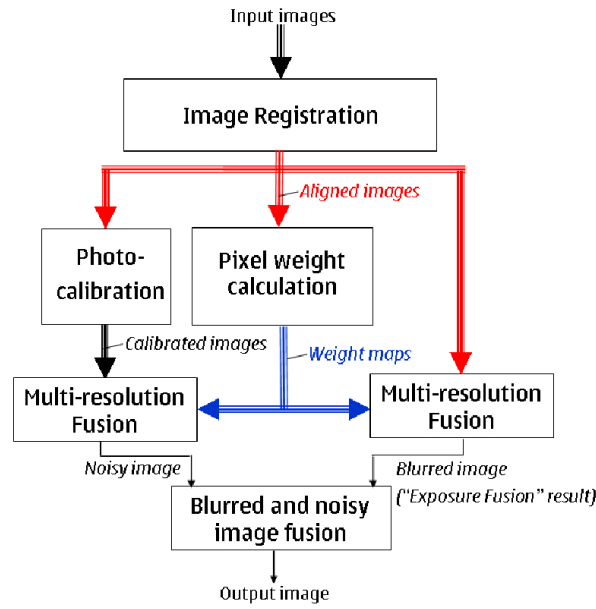
**FIGURE 3.3** – Scènes de grande gamme dynamique capturées grâce au capteur logarithmique de Yang Ni et al. [77].

#### 5. Capture multiple & conclusion

Comme montré dans la Section 2.3, la capture multiple utilise plusieurs images prises avec des temps d'exposition différents afin de synthétiser une nouvelle image, dite HDR, où tous les détails des parties partiellement sombres et lumineuses d'une scène sont bien représentés. D'un point de vue qualitatif, il a été démontré que cette technique a le meilleur rapport signal sur bruit [78], comparée aux techniques citées ci-dessus. C'est aussi une méthode largement utilisée notamment dans la photographie amateur. La dynamique peut théoriquement être augmentée sans limites, en choisissant un nombre d'images approprié à la dynamique de la scène visionnée. Toutefois, cette technique présente des inconvénients, tels que la nécessité d'avoir un traitement post-capture assez lourd, avec des contraintes de timing qui ne sont pas faciles à respecter, plus spécialement dans un contexte de capture vidéo. Cette technique n'augmente pas la dynamique propre du capteur, mais augmente plutôt la quantité d'information capturée et perçue au final par l'utilisateur. Dans l'attente de nouvelles technologies révolutionnaires, c'est la façon de faire la plus usitée de nos jours dans le domaine du HDR. Nous allons voir, dans les sections suivantes, les architectures logicielles et les systèmes matériels existants qui se sont majoritairement appuyés de cette technique.

### 3.3/ LES SYSTÈMES À FUSION D'EXPOSITION

Bachoo [79] a mis au point une application dédiée à la technique de fusion d'exposition en 2009, afin de fusionner en temps réel des vidéos à trois niveaux de gris, de résolution  $1600 \times 1200$  et à la cadence de 20 images par seconde. Il est capable de contrôler la vitesse de génération d'images, en ayant un framerate constant, relatif à une taille définie des blocs de traitement et à une résolution de la vidéo fixée. Il effectue l'implantation d'une variante de l'algorithme de Goshtasby [65] décrit dans la Section 2.3.2. L'implémentation est faite sur CPU et sur GPU. L'algorithme est divisé en deux parties de telle sorte que les puissances de traitement du CPU et du GPU ("Graphics Processing Unit") sont utilisées judicieusement. La capacité du CPU permet d'effectuer les opérations massivement séquentielles telles que le calcul des entropies. À contrario, le GPU sert à fusionner les blocs entre eux, opération qui peut être parallélisée afin d'augmenter la vitesse d'exécution du processus de fusion. La vitesse de défilement du rendu final est de 20 images par seconde, mais elle peut augmenter si la résolution vidéo diminue ou si la taille des blocs de traitement augmente. En jouant à la fois sur la résolution des vidéos et la taille des blocs de traitement, on peut ainsi se fixer un framerate adapté, un compromis entre vitesse d'exécution et qualité du traitement. Rien n'est dit quant au choix des temps d'exposition et aucune méthode n'est proposée pour estimer les expositions nécessaires au bon fonctionnement de leur système. Il est indiqué que l'utilisation d'expositions LDRs supplémentaires peut entraîner un goulot d'étranglement dans le processus de fusion des images.



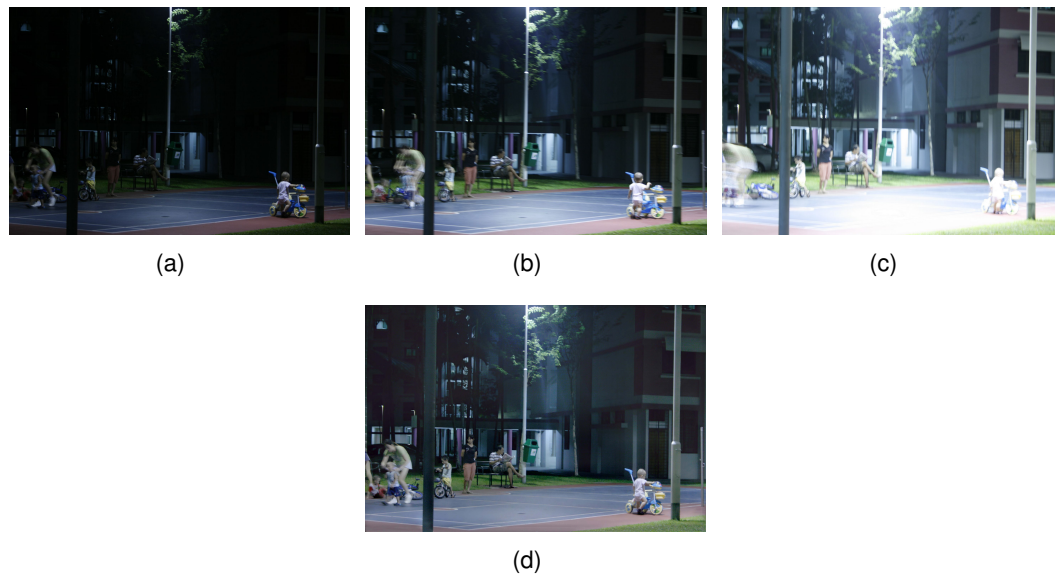
**FIGURE 3.4** – Le schéma fonctionnel de la méthode proposée par Tico et al. [80].

Les images capturées avec des temps d'intégration faibles sont souvent affectées par le bruit du capteur. Au contraire, une image prise avec un long temps d'exposition est moins bruitée, mais est altérée par le flou généré par le mouvement local dans la scène, et le flou global qui est dû à la caméra. Une technique de fusion de deux expositions a été élaborée par Tico et al. [80] pour surmonter ces limites, convenant à un appareil mobile. L'exposition la plus courte est utilisée comme une référence avec le moins d'effets indésirables liés au mouvement, et la plus longue exposition est utilisée comme l'image de référence avec un bruit faible. Le schéma du système complet est visible sur la Figure 3.4. Une calibration photométrique est faite, représentée par la fonction de transfert discrète de luminosité ( $BTf(i)$ ) entre les deux images. Cette fonction s'appuie sur l'utilisation de la technique d'histogramme conjoint, par la formule suivante :

$$C(i, j) = \sum_x \omega(i)\omega(j)\delta[I_1(x) - i]\delta[I_2(x) - j], \quad (3.3)$$

où  $\delta$  est la fonction Dirac,  $x$  est l'indice de coordonnée du pixels dans l'image,  $i$  et  $j$  sont les niveaux de luminosité dans les images  $I_1$  et  $I_2$ , et  $\omega$  est une fonction de pondération qui donne un poids plus élevé au pixel proche du milieu de la dynamique du capteur (proche de la valeur 128 pour un capteur 8 bits par exemple).  $BTf(i)$  donne à chaque valeur  $i$ , la valeur de luminosité  $j$  la plus probable pour l'autre image. On obtient ainsi une image  $I_N$  à basse exposition, et dépourvue d'effet de flou. Puis, ils fusionnent  $I_N$  avec le résultat de la fusion de l'exposition  $I_b$ . Le résultat donne une image qui n'est ni affectée par le flou de mouvement (comme on peut le voir sur la figure 3.5), ni par le bruit du capteur (à faible exposition). On remarque tout de même que les détails dans les zones

à faibles luminosités ne sont pas très bien visibles (fenêtres de l'immeuble).



**FIGURE 3.5** – Résultat du système initié par Tico et al. [80]. (a), (b) et (c) sont l'ensemble des expositions simples LDRs, et (d) est le résultat de leur méthode.

Le document de Youm et al. [81] présente une méthode pour générer une vidéo HDR grâce à la fusion de trames vidéos d'expositions contrôlées. Deux questions doivent être examinées : le contrôle de l'exposition et la gestion des objets en mouvement avec la méthode d'ajustement du masque. Le document présente de façon précise l'algorithme de contrôle d'exposition, et l'algorithme de combinaison de deux images consécutives, utilisant l'approche basée sur la fusion de pyramide. L'avantage de la méthode est de sélectionner automatiquement les temps d'exposition appropriés et de réduire la distorsion du flou de mouvement. L'algorithme de contrôle de l'exposition détermine des masques de régions d'intérêt (ROIs) pour les régions claires et sombres de la scène. Ensuite, ils appliquent la sélection d'exposition à ces masques. Pour étendre la gamme dynamique, ils utilisent l'approche pyramidale en combinant les trames vidéos voisines dans le temps. Pour réduire les artefacts dus au déplacement des objets, on ajuste le masque des régions de saturation en utilisant le masque de variation entre les régions. Ce masque est obtenu en détectant les variations entre les expositions (longues ou courtes) courantes et précédentes. L'algorithme proposé donne des résultats qui ont plus de visibilité et plus de détails dans un environnement à l'intérieur qu'à l'extérieur. Néanmoins, il y a présence d'effets de "ghost" si de grands mouvements sont impliqués. Malheureusement, les traitements ne sont pas réalisés en temps réel, puisqu'il faut  $2.5s$  pour générer chaque trame vidéo HDR.

### 3.4/ LES SYSTÈMES DE CAPTURE

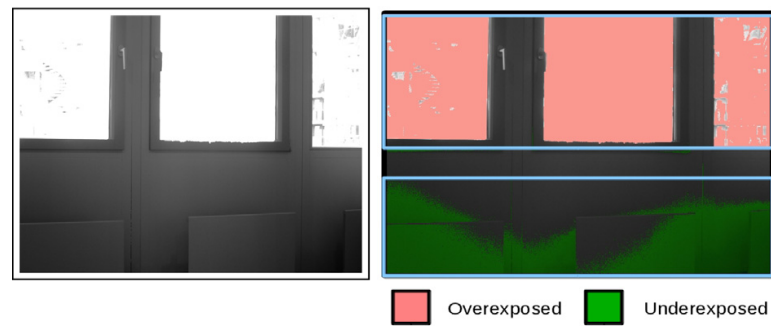
Tocci et al. [82] élaborent en 2011, une nouvelle architecture optique capable de capturer plus de 17 stops de la plage dynamique (environ  $100dB$ ). Le système est construit autour de trois capteurs, qui peuvent saisir simultanément des images hautes, moyennes et basses expositions. Ils utilisent des séparateurs de faisceau lumineux entre l'objectif et les capteurs, afin de réfléchir partiellement et de répartir la lumière sur les trois capteurs. La fusion des images est implémentée sous Matlab. Ils utilisent les valeurs de pixels issues du capteur avec la plus longue exposition (donc le moins bruité), pour surmonter les problèmes de quantification identifiés dans l'algorithme de Debevec et de Malik [10]. En fait, les artefacts se produisent lorsque les images LDRs sont séparées par plus de trois stops photographiques. Le post-traitement est composé des algorithmes de fusion HDR et de Tone Mapping. Le premier est mis en œuvre sous Matlab, le second se fait avec un outil de traitement HDR en software, à l'aide d'un logiciel tel que Photomatix. Dans le système ainsi élaboré, seule la partie capture traite des considérations vidéo en temps réel.

L'article de Guthier et al. [83] a introduit en 2012 un algorithme où une meilleure qualité HDR peut être réalisée avec le même nombre de captures LDRs. Le choix des expositions est effectué de façon optimale, en sélectionnant les "vitesses d'obturation" qui contribueront le plus efficacement à rajouter de l'information utile à l'image HDR finale. Dans un contexte pouvant être temps réel, minimiser le nombre de captures est primordial pour accélérer le processus de traitement. Fondamentalement, les temps d'exposition sont choisis de telle sorte que les valeurs de luminosités enregistrées à une position  $i, j$ , correspondent à une exposition correcte dans au moins une des images LDRs capturées. Tout d'abord, une bonne approximation de la valeur de radiance  $E$  est calculée en tenant compte de la fonction de réponse  $f$  de l'appareil de capture, de la vitesse d'obturation  $\Delta t$  et de la valeur du pixel. Une fonction de pondération juge, en fonction des valeurs de luminosités, de l'utilité du pixel pour la détermination de la valeur de radiance finale. La fonction de contribution  $c$  pour chaque position  $(i, j)$  est alors définie par :

$$c_{\Delta t}(E) = \omega(f(E\Delta t)) \quad (3.4)$$

L'histogramme des radiances peut être utilisé pour calculer une séquence de vitesses d'obturation en choisissant  $\Delta t_i$  telle que les pics de la fonction de contribution  $c_{\Delta t}(E)$  de l'image LDR coïncident avec les pics de l'histogramme des radiances. Une relation utile est donc établie entre l'histogramme de radiance et le vecteur de contribution qui indiquerait potentiellement que le changement, et non la forme de la fonction de la contribution, dépendrait de la vitesse d'obturation dans le domaine logarithmique. Pour calculer la séquence des temps d'exposition optimaux, le vecteur de contribution initial doit être calculé en premier, et la vitesse d'obturation initiale  $\Delta t$  peut être choisie arbitrairement pour calculer  $c_{\Delta t}$ . Le vecteur de contribution est déplacé dans l'histogramme de radiance dans



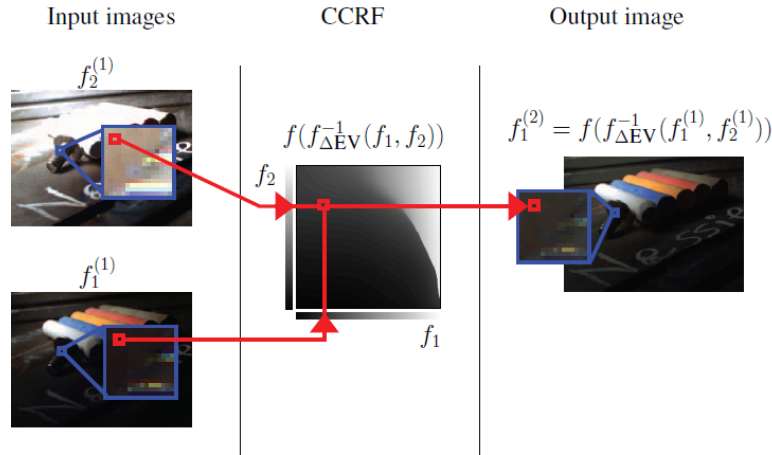


**FIGURE 3.6** – La ré-exposition partielle par Guthier et al. [84]. Les zones surexposées et sous-exposées sont rassemblées en blocs ROI (cadres bleutés) afin d’être re-capturées. La durée normale d’exposition multiple est ainsi réduite.

lequel chaque position correspond à une vitesse d’obturation  $c_{\Delta t_i}$ . A partir de là, on peut choisir d’ajouter ou non une exposition supplémentaire, et de sélectionner les nouveaux temps d’exposition ayant le plus grand gain en termes d’informations apportées à l’image HDR. Pour cela, le vecteur de contribution combiné  $C(E_j)$  est défini. Avec l’hypothèse selon laquelle la qualité dépend de la valeur de la plus forte contribution qu’apportent les expositions, le temps d’exposition correspondant au décalage qui conduit à la plus grande augmentation de  $C$  est ajouté à la séquence. Néanmoins, il y a deux problèmes majeurs qui se posent lorsque l’on applique cet algorithme : l’imperfection du calcul d’histogramme et le flicker. Un critère de stabilité est également introduit à la séquence d’obturation qui permet éventuellement à chaque trame d’être ajustée jusqu’à ce qu’une séquence d’obturation stable soit trouvée. La technique habituelle de capture, à savoir choisir des expositions qui sont régulièrement réparties dans la gamme dynamique, gaspille trop de temps à la capture et ne contribue pas assez à l’apport judicieux d’information durant la reconstruction de la carte de radiance. Au final, grâce à cet algorithme, on économise le temps de capture tout en étant capable de réduire le nombre d’expositions LDRs sans perte de qualité au final.

Une autre revue de Guthier et al. [84], parle d’une approche de capture qui réduit la quantité de données capturées en ne re-capturant que les mauvaises zones mal exposées. Cela nécessite plusieurs étapes. Premièrement, une image en pleine résolution est capturée. Puis celle-ci est analysée en marquant les lignes de l’image où l’histogramme montre des parties exposées de façon incorrecte. Après le regroupement de ces lignes en simples ROIs, la recapture est effectuée selon le principe de dichotomie, jusqu’à ce que toutes les zones de l’image soit bien exposées. L’illustration de cette méthode de capture est montrée dans la figure 3.6. En fait, le but est d’éviter de réexposer l’image entièrement, ce qui diminue le temps de capture. Ainsi, 29% de la durée totale de capture peut être sauvée.





**FIGURE 3.7** – Utilisation d'une CCRF ("Compparametric Camera Response Function") par Ali et al. [85]. L'estimation de la radiance est donnée par  $\hat{q} = f_{\Delta EV}^{-1}(f_1, f_2)$ , où  $\hat{q}$  est la valeur de radiance,  $f$  l'estimateur commun où chaque estimation de radiance dépend de plusieurs mesures.

### 3.5/ LES SYSTÈMES DE CRÉATION DE CONTENU HDR

Une méthode basée sur les tables LUTs ("Look-Up Tables") pour la génération HDR a été introduite en 2012 par Ali et al. [85]. Ils développent des algorithmes pour reconstruire une image HDR avec l'utilisation d'une fonction "compparametric" de réponse de la caméra (CCPR). La capture des images avec un intervalle constant de valeur d'exposition est proposée. La différence d'exposition est notée  $\Delta EV$ . Ils calculent préalablement une LUT multidimensionnelle pour la fonction CCPR de taille  $1024 \times 1024$ . La taille de la LUT est choisie de manière empirique et dépend de la plage d'exposition utilisée.

$$\hat{q}_i = f_{\Delta EV}^{-1}(f_i, f_{i+1}), i \in \{1, \dots, N\}, \quad (3.5)$$

où  $f_i$  est la valeur du pixel retournée par la caméra dans l'image  $i$ . Construire une LUT CCPR pour  $f_1$  et  $f_2$  signifie que l'on peut exécuter toutes les estimations  $\hat{q}$  pour toutes les combinaisons de  $f_1, f_2 \in \{0, \frac{1}{1023}, \frac{2}{1023}, \dots, 1\}$ . Une illustration de l'utilisation de cette CCRF est donnée dans la figure 3.7. Une extension de la composition "compparametric" de l'image se décline à la fois pour un nombre d'images différent (3 et plus), pour différentes valeurs d'écarts d'exposition  $\Delta EV$  et pour une topologie de connexion différente, et ce, quelle que soit l'application. Les méthodes sont mises en œuvre sur CPU et GPU. Par rapport à la méthode de calcul directe conventionnelle, la vitesse de calcul d'un pixel HDR est multipliée par 3800 avec une mise en œuvre sur GPU, pour une sortie cadencée à 398 images par seconde. Les auteurs mentionnent que la plupart du temps utilisé est le temps passé à attendre le transfert de données entre l'hôte et le GPU. Le principal défaut de cette méthode est l'utilisation de temps d'exposition fixes. Tout changement de luminosité dans la scène capturée impliquerait nécessairement le calcul de nouvelles LUTs multidimensionnelles correspondantes aux nouveaux temps d'exposition. La mé-

thode semble donc n'être applicable que dans des conditions de luminosité connues et gérables.

Castro et al. [16] ont présenté une méthode pour la génération de vidéo High Dynamic Range. La solution est mise en œuvre dans un smartphone Nokia N900. La reconstruction de vidéo HDR a besoin à la fois de matériel comme une caméra programmable, et également de développement logiciel. Les phases critiques du pipeline HDR, telles que la récupération de la carte de radiance ou l'alignement des images, sont basées essentiellement sur les histogrammes pour un faible coût en termes de ressource. Les méthodes implantées sont les suivantes :

- Récupération de la fonction de réponse de la caméra [86] [12]
- Algorithme de reconstruction HDR [87]
- Alignement des images par une technique basée sur les histogrammes [68]
- Reconstruction de la carte de radiance avec la suppression de "ghost" [71]

Tout d'abord, une pyramide d'images est construite à partir de chaque image en niveau de gris. Ensuite, pour chaque niveau dans la pyramide, une image bitmap MTB (Median Threshold Bitmap) est construite à partir d'un histogramme. La méthode utilisée correspond à celle initiée par Ward [68] (voir Section 2.4.1).

Les effets de "ghost" sont traités par l'analyse de la variance des valeurs de radiance sur les pixels des images alignés  $F^i$ . Pour y arriver, quatre images HDRs sont reconstruites, une pour chaque  $F^i$  et une à partir de toutes les images. Pendant que l'utilisateur regarde dans la visionneuse (sans enregistrement), l'application exécute un algorithme d'exposition automatique, corrige la balance des blancs et affiche un aperçu de la scène en utilisant les paramètres actuels. Lorsque la capture démarre, l'algorithme d'exposition automatique est exécuté en arrière-plan, et fournit des paramètres optimaux dans le but d'avoir une bonne exposition tout au long du processus de capture vidéo. La première capture utilise une exposition optimale pré-calculée de valeur  $x$ , la deuxième et la troisième exposition sont respectivement  $2x$  et  $\frac{x}{2}$ . Ce processus est répété jusqu'à ce que la vidéo soit entièrement enregistrée et que les images soit envoyées à l'étage de traitement. Toutes les captures utilisent le même gain afin de minimiser le bruit et de garantir que la courbe de réponse du système ne varie pas. La mise en œuvre actuelle effectue cette étape sur un ordinateur de bureau à des fins de test.

### 3.6/ LES SYSTÈMES À REPRODUCTION DE TONS

En 2007, Hassan et al. [88] ont décrit une architecture basée sur FPGA, réalisant le tone mapping d'images HDRs en niveau de gris. Ils sont capables de compresser la gamme dynamique des images HDRs de résolution  $1024 \times 768$ , à la cadence de 60 images/s. Le système est basé sur une modification de l'algorithme local de Reinhard et al. [14]. Plusieurs limites, néanmoins intéressantes, découlent de leurs travaux. D'une part, les



**FIGURE 3.8** – Le célèbre mémorial par Debevec et al. [89], utilisé comme image HDR de référence dans le domaine du HDR, au travers de nombreux systèmes logiciels existants. Il est ici tone mappé pour pouvoir l'afficher sur écran standard.

auteurs se focalisent uniquement sur l'opération de tone mapping, et laissent de côté l'aspect capture et génération de carte de radiance. Ils utilisent donc une image HDR de test, celle du mémorial tirée du travail de Debevec et al. [89] (voir Figure 3.8). L'image HDR du mémorial est une référence commune pour les auteurs du domaine HDR. Ils l'utilisent afin d'évaluer leurs algorithmes de tone mapping et peuvent ainsi comparer leur algorithme avec d'autres algorithmes. D'autre part, les TMOs ont souvent besoin d'avoir accès à des informations qui sont par définition dépassées dans le temps, calculées une fois que la totalité d'une image HDR a été générée. L'utilisation des valeurs de radiance minimum, maximum ou moyenne sont nécessairement calculées vis à vis de la trame précédente. Dans le cas d'une application vidéo, la méthode préconisée est de calculer ces paramètres à partir de l'image HDR précédemment générée, et d'appliquer ces paramètres lors du tone mapping de l'image HDR courante. Enfin, l'utilisation de la pyramide gaussienne afin d'effectuer le tone mapping local sur l'image HDR nécessite une utilisation importante de la mémoire externe. La suite de ce travail, continué au travers de la thèse de Vytla en 2010 [90], a été l'implémentation de l'algorithme de Fattal et al. [17], moins complexe et moins gourmand en mémoire que celui de Reinhard. La clé de cette implantation est le design d'une solution de résolution de l'équation de Poisson simplifiée. En théorie, cette implantation temps réel de l'algorithme de Fattal et al. permet le tone mapping d'image HDR de 1 Mégapixel à 100 fps. Seules des simulations et des résultats de synthèse sont présentés.

Certains opérateurs de tone mapping ont été adaptés pour être applicable en vidéo HDR. En 2012, Kiser et al. [91] présentaient une solution complète de tone mapping qui peut être mise en œuvre en temps réel. Ils utilisent une version modifiée de l'opérateur global

de Reinhard et al. [14]. Le paramètre "key" (appelé  $a$ ) devient dynamique, en fonction des luminances minimale, maximale et moyenne  $L_{min}$ ,  $L_{max}$  et  $L_{av}$  calculées à partir de la carte de radiance de l'image HDR produite précédemment :

$$a = 0.18 * 2^{2(B-A)/(A+B)}$$

$$A = L_{max} - L_{av} \quad (3.6)$$

$$B = L_{av} - L_{min}$$

Puis, l'algorithme photographique de reproduction de tons est appliqué normalement. La méthode a l'avantage de réduire le scintillement vidéo appelé aussi "flicker", grâce à l'utilisation d'un intégrateur à fuite, que nous ne détaillerons pas, appliqué à l'équation 3.6. Leur système est indépendant des paramètres d'image en entrée, et est implémentable en temps réel pour s'adapter au débit de données vidéos.

Ching-Te et al. [92] ont défini une méthodologie afin de développer un processeur de tone mapping optimisé, utilisant une plateforme ARM SOC (System On Chip). Ils utilisent leur processeur afin d'évaluer à la fois les méthodes de compression photographique de Reinhard [14] et de compression de gradient de Fattal [17], et ceci pour différentes applications. Ce nouveau processeur peut compresser des images HDR de résolution  $1024 \times 768$ , à la cadence de 60 fps. Le cœur du processeur a besoin d'une superficie matérielle de  $8.1mm^2$  avec la technologie TSMC  $0.13\mu m$ .

### 3.7/ LES SYSTÈMES HARDWARE

#### 3.7.1/ LES SYSTÈMES NON TEMPS RÉEL

Kang et al. [93] décrivent en 2003 une solution algorithmique qui permet d'effectuer une capture vidéo et la synthèse HDR. Cette revue est une des plus citées dans le domaine de la vidéo HDR, et a notamment fait l'objet d'un dépôt de brevet. La solution est capable de générer une vidéo HDR à partir d'une séquence de deux images d'une scène capturée, en faisant varier l'exposition de chaque image (alternant les expositions claires et sombres). A cet effet, l'approche se compose de plusieurs parties principales :

- La capture d'une séquence d'images alternant basse et haute expositions
- L'identification des pixels entre l'image précédente et ses deux images voisines
- La définition d'un vecteur de mouvement attribué à chaque pixel
- La fusion bidirectionnelle des images voisines par un procédé ingénieux de "boost" des intensités de l'exposition basse voisine, afin de faire correspondre cette image avec l'image à haute exposition
- Le tone mapping qui réduit la dynamique des images HDR générées.

La méthode permet de supprimer les effets de ghost, apparaissant à la fois localement et globalement, grâce aux correspondances faites entre les images adjacentes. Cependant, l'utilisation software des algorithmes via un ordinateur muni d'un Pentium 4.2GHz n'atteint pas la contrainte de temps réel, car le temps de traitement pour chaque trame vidéo ( $1024 \times 768$ ) est d'environ 10 secondes (8 secondes pour la cartographie des luminances et 2 secondes pour le tone mapping). Basé sur ces algorithmes, Youm et al. [81] ont créé une vidéo HDR en fusionnant deux images de différentes expositions acquises par un système de caméra vidéo fixe. Leur méthodologie repose essentiellement sur la simple stratégie du contrôle automatique des temps d'exposition et de leur combinaison de manière efficace. Malheureusement, ils n'atteignent pas non plus un système de traitement en temps réel, avec un temps d'environ 2,5 secondes pour chaque image de ( $640 \times 480$ ) pixels sur un processeur d'1,53 GHz AMD Athlon XP 1800+.

En 2010, Gelfand et al. [94] ont proposé un système qui fournit comme fonctionnalité une visionneuse, la capture et la fusion des images sur une caméra de type téléphone programmable. La contribution innovante de leurs travaux est qu'ils ne capturent pas un nombre prédéterminé d'images LDRs pour la reconstruction HDR. Basé sur le calcul de la dynamique de la scène, fait à partir de deux images capturées avec deux temps d'exposition extrêmes, ils en déduisent le nombre d'images devant être acquises dans le but de récupérer toute la dynamique de luminances de la scène observée. Ils exigent que moins de 10% des pixels dans l'exposition extrême courte soient saturés, et exigent que moins de 10% des pixels dans l'exposition extrême longue aient des valeurs inférieures à 16. Lorsque les deux expositions sont stables, la mesure est jugée bonne. Cette fonction est alors appliquée :

$$D_y = \frac{T_L \times G_L}{T_S \times G_S}, \quad (3.7)$$

où  $D_y$  est l'estimation de la dynamique de la scène,  $T_S$  et  $G_S$  sont le temps d'exposition court et le paramètre de gain pour l'exposition courte. De la même manière  $T_L$  et  $G_L$  sont les paramètres de l'exposition longue. Puis, selon la valeur de  $D_y$  obtenue, ils en déduisent si une, deux ou trois expositions sont nécessaires à la reconstruction de l'image HDR. La fusion des images se fait avec une version modifiée de la fusion d'exposition initiée par Mertens et al. [66]. De leur point de vue, cet algorithme est une approche intéressante pour la production d'images HDR-like sur un téléphone mobile, puisque tous les calculs sont effectués en codage 8-bit dans le domaine de l'image. Malheureusement, la gestion mémoire et les calculs fastidieux des cartes de pondération nécessaires à la fusion, ne leur permettent pas de travailler au sein d'un système en temps réel. Cela ne leur permet donc pas d'obtenir un résultat directement affichable.

Unger et al. [95] ont développé un nouveau système de vision capable de capturer des images couleurs à la cadence de 25 images/s à l'aide d'un mode "rolling shutter". Partant du constat qu'ils utilisent une grande vitesse de capture et un accès direct au convertisseur Analogique/Digital linéaire, ils se permettent de ne pas utiliser de correction de mouvement dans la scène. Ils ne jugent pas bon de se servir de méthodes comme le

moyennage pour réduire les artefacts. De plus, ils ont la possibilité d'ajouter deux bits supplémentaires de précision (malgré l'augmentation du bruit) pour les convertisseurs A/D 8 – *bit*, en faisant varier le gain des amplificateurs. Huit "readout" sont effectués, mais seulement cinq expositions sont réellement acquises. Les trois autres "readouts" sont des extensions en gain des trois expositions les plus longues. Ce système permet une résolution maximale de 512 lignes, et un framerate maximum de 25 fps. Ils prennent la valeur maximale des pixels non-saturés parmi les expositions disponibles. L'encodage en format flottant est réalisé de façon linéaire puisque telle est la particularité de leurs convertisseurs. Cette formule est donc employée :

$$E = \frac{x_i}{\Delta t_i}, \quad (3.8)$$

où  $x_i$  est la valeur du pixel issue du capteur,  $E$  est l'irradiance et  $\Delta t_i$  est le temps d'exposition du pixel dans l'image  $i$ . Afin d'avoir la meilleure précision, les autres pixels saturés ne sont pas utilisés. Le système est une technologie très mature, avec un capteur d'une bonne sensibilité à la lumière, avec peu d'effet de "blooming", et facilement calibrable. En outre, l'accès direct au convertisseur linéaire A/D ne nécessite pas la bonne connaissance de la courbe de réponse du système. La gamme dynamique de leur solution va jusqu'à 10,000,000 : 1, avec une exposition très courte de l'ordre de  $1\mu s$ . Malheureusement, une optique de haute qualité est nécessaire et le capteur doit être refroidi pour une plage dynamique extrême, sinon la qualité d'image peut être affectée par le bruit. En raison du mode de capture en mode "rolling shutter" et d'une grande variation dans les temps d'exposition, des artefacts peuvent aussi apparaître lors de la présence de mouvement dans la scène, et peuvent provoquer des erreurs sur les valeurs de pixels.

### 3.7.2/ LES SYSTÈMES TEMPS RÉELS

Le document de Ureña et al. [96] publié en 2012, a présenté deux architectures temps réel, décrites à la fois sur GPU et sur FPGA. Les implémentations ont été faites sur un appareil portatif fonctionnant sur batterie. Une nouvelle génération de tone mapping est présentée dans cet article. L'innovation vient du fait d'une conception peu coûteuse en ressources, puisque la plupart des opérateurs existants nécessitent des opérations coûteuses en temps de calcul ou en mémoire, rendant ces opérateurs impossible à mettre en œuvre sur système matériel. L'opérateur est construit de telle manière à atténuer les éclats dans les images pour les personnes malvoyantes. L'opérateur de tone mapping comprend à la fois des approches de calcul globales et locales. Typiquement, pour l'aspect global, il renforce les zones à faibles contrastes de luminosité, mais peut aussi bien préserver les régions où le contraste est satisfaisant. Localement, il atténue les régions trop vives ou bien renforce les éclats afin d'améliorer les détails de l'image. L'amélioration globale est basée sur l'adaptation des histogrammes de luminosités par canal, dans l'espace de couleurs HSV. D'autre part, la mise en valeur locale est fondée sur un schéma

de traitement de type "rétina-like". Grâce à l'utilisation d'un canal RVB en entrée, le nouvel opérateur a la possibilité de renforcer le contraste des couleurs qui s'opposent, afin d'améliorer les contours de la scène et d'ajuster le processus automatiquement en fonction des conditions d'éclairage. Pour résumer, les filtres gaussiens, les facteurs de pondération et la prise en compte du système visuel humain sont les principaux avantages de l'opérateur. Voici quelques caractéristiques du système ainsi établi :

- Réduction des temps de calcul pour le temps réel
- Ajustement automatique des paramètres en fonction des conditions d'illumination
- Accentuation des contours pour les personnes mal voyantes
- Atténuation des éclats
- Fourniture des détails de qualité meilleure dans les zones fortement éclairées
- Non destruction de la cohérence des couleurs dans l'image

Le Tableau 3.1 résume les comparaisons en utilisant les deux plates-formes de test différentes. L'implémentation FPGA produit une vidéo avec un framerate élevé, consommant peu de puissance électrique, alors que l'implémentation GPU fournit plus de précision dans le calcul des pixels HDRs, mais est plus gourmand en ressources. Les deux systèmes peuvent être réduits à une application portable fonctionnant sur piles ou batteries.

<i>Paramètre</i>	<i>GPU</i>	<i>FPGA</i>
Framerate	Limité (30fps)	Élevé (60fps)
Gamme dynamique de calcul	Flottant	Virgule fixe
Précision	↑	↓
Puissance utilisée	↑	↓
Architecture	Fixe	Flexible
Mise en œuvre	Aisée	Difficile

**TABLE 3.1** – Résumé de comparaisons résultant des implémentations de la méthode d'Ureña et al. [96] sur deux architectures différentes.

En 2012, A. O. Akyüz [97] développe un système complet sur plate-forme GPU. Les tâches sont effectuées en parallèle et en structure de type pipeline. La génération HDR et le tone mapping se font sans recourir à la récupération de la courbe de réponse de la caméra. Ils utilisent la formule initialement proposée par Debevec et al. [10] pour calculer les valeurs de radiances. En ce qui concerne le fonctionnement du tone mapping, c'est l'algorithme de Reinhard et al. [14] qui a été choisi et mis en place. Certains résultats sont identiques aux méthodes de mise en œuvre sur CPU. Ils atteignent une cadence de 65 fps pour la production d'images HDRs de 18Mpixels, et 103 images par seconde pour le tone mapping local. Le matériel utilisé est une carte Nvidia GeForce GTX 590. Toutefois, dans leur calcul de performances, ils ne prennent pas en compte le temps de chargement des textures sur le GPU. Or la majorité du temps est consacré au transfert des pixels des images LDR d'origine vers le GPU. Les calculs de luminances et de poids n'ont montré qu'un impact faible sur la vitesse de calcul, et donc sur la cadence du système final.

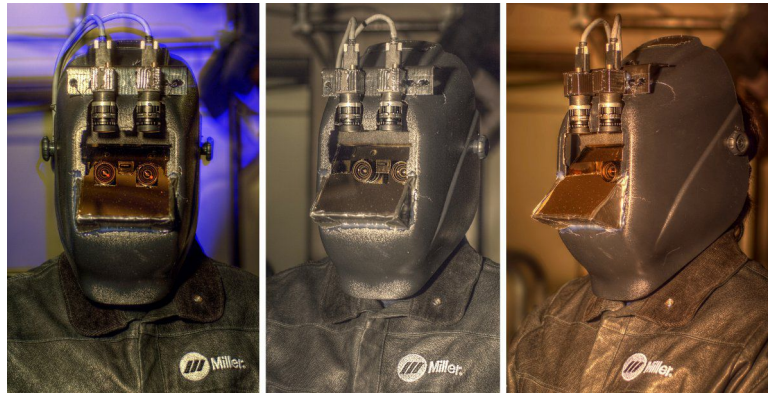
Ke et al. [98] proposent en 2009 une méthode innovante pour générer des vidéos HDRs affichables sur un écran classique. Cette méthode diffère radicalement des méthodes de l'état de l'art, puisqu'une seule image LDR est suffisante pour générer des images HDRs virtuelles. Les détails fins et uniformément répartis en intensité sont représentés, accentuant ainsi l'impression subjective de qualité d'image, mais n'augmentant en aucun cas la quantité d'information présente sur l'image HDR résultante. Ils mettent en œuvre un synthétiseur d'images HDR hardware efficace, qui comprend entre autres une amélioration du contraste local. Ce système a une complexité beaucoup plus faible par rapport aux systèmes classiques. Avec la technologie *UMC90nm*, ils réalisent une vidéo en temps réel à 60 fps et à une résolution de  $720 \times 480$  pixels.

Enfin, l'un des systèmes complets des plus populaires est basé sur le système Mann Vision [99] (du nom de son inventeur, Steve Mann, voir Figure 3.9). C'est donc en 2012 qu'un casque de soudage arborant deux caméras vidéo commandées par ordinateur a été présenté. Les données reçues par ces caméras sont enregistrées ligne par ligne dans une mémoire externe de type SDRAM. Plusieurs FIFOs bufferisent les pixels et les relisent simultanément ligne par ligne. Le nombre de FIFOs dépend du nombre d'images utilisées lors de la reconstruction des images HDRs. Une LUT contenant des valeurs pré-calculées est utilisée pour combiner les expositions multiples. Cette LUT est inspirée du travail d'Ali et al. [85], sur l'estimation de radiance avec l'aide d'une CCRF ("Comparametric Camera Response Function") pré-calculée. Pour le cas d'une combinaison de deux images couleurs  $f_1$  et  $f_2$ , une LUT de taille  $256 \times 256 \times 3$  doit être calculée. L'estimation de la LUT  $Q_f(f_1, f_2)$  est faite en calculant toutes les valeurs possibles en sortie pour chaque combinaison de pixels de  $f_1$  et  $f_2$ . Les paramètres intervenant dans le calcul de cette LUT sont l'écart d'exposition entre les images  $\Delta EV$ , la fonction de tone mapping  $c$ , la courbe de réponse du système  $f$  et un paramètre de contraste  $k$ . Ainsi :

$$Q_f = k(c(f_{\Delta EV}^{-1}(f_1, f_2))) = C_{LUT}(f_1, f_2) \quad (3.9)$$

La technique est extensible à une architecture utilisant plus d'images, sans en augmenter significativement la complexité. Avec cette méthode et grâce à l'utilisation massive de LUTs, ils ont la possibilité d'obtenir une vidéo avec un temps de calcul fixe et contrôlé (autrement dit temps réel) sur un FPGA Spartan-6 LX45. Le problème de cette méthode est qu'ils sont obligés d'utiliser des valeurs d'exposition prédéterminées une fois le calcul des LUTs effectué, ce qui semble rendre le système inutilisable lorsque les conditions de luminosités changent. Néanmoins, cette méthode peut tout à fait convenir à leur application visée, à savoir la soudure en milieu industriel, dans des conditions lumineuses fixées.





**FIGURE 3.9** – Le système de casque de soudage "MannVis welding helmet" par Mann et al. [99].

### 3.8/ CHOIX MATÉRIEL POUR UN SYSTÈME DE VISION HDR EN TEMPS RÉEL

Une fois l'état de l'art des systèmes existants réalisé, nous pouvons maintenant sélectionner le matériel qui va nous servir à concevoir notre système vidéo temps réel. Nous allons expliquer nos choix ici. Il a fallu définir les contraintes techniques réelles liées au contexte de la thèse, se référer au matériel mis à disposition au laboratoire, envisager de réutiliser les connaissances issues du laboratoire dans tel ou tel domaine, etc.

#### 3.8.1/ LE CAPTEUR

Le choix de notre capteur s'est fait assez rapidement, du fait du contexte de la thèse. e2v est une entreprise faisant partie, comme le laboratoire LE2I, du projet européen Hidralon. Ce projet s'incluant dans la recherche de nouvelles solutions de fabrication des capteurs "Low Noise", e2v nous a fourni des échantillons de capteur couleurs et N&B, ainsi qu'un kit de démonstration. Le capteur était destiné à être commercialisé, seulement les échantillons que nous avions étaient des versions Beta. Nous avons donc eu le privilège de tester le capteur avant même sa commercialisation. Notre système de vision a été construit autour de ce capteur disponible en tout début de thèse.

Le capteur EV76C560 d'e2v [101] est issu de la technologie CMOS, et est doté d'une résolution de 1,3 million de pixels ( $1280 \times 1024$ ) cadencée à 60 ips à pleine résolution. Il offre une gamme dynamique de sortie de 10-bit en lecture numérique. Il embarque des fonctionnalités telles que la possibilité d'avoir l'histogramme d'image de sortie, le multiple ROIs ("Region Of Interest"), la correction de pixel défectueux, le global shutter, etc. Il y a aussi possibilité de configurer une séquence d'acquisition d'images successives avec des temps d'exposition différents (jusqu'à quatre temps). Grâce à cette fonctionnalité, on peut alors s'affranchir de la reconfiguration des registres du capteur entre chaque image. Il a aussi la particularité d'avoir une bonne sensibilité à des niveaux de luminosité

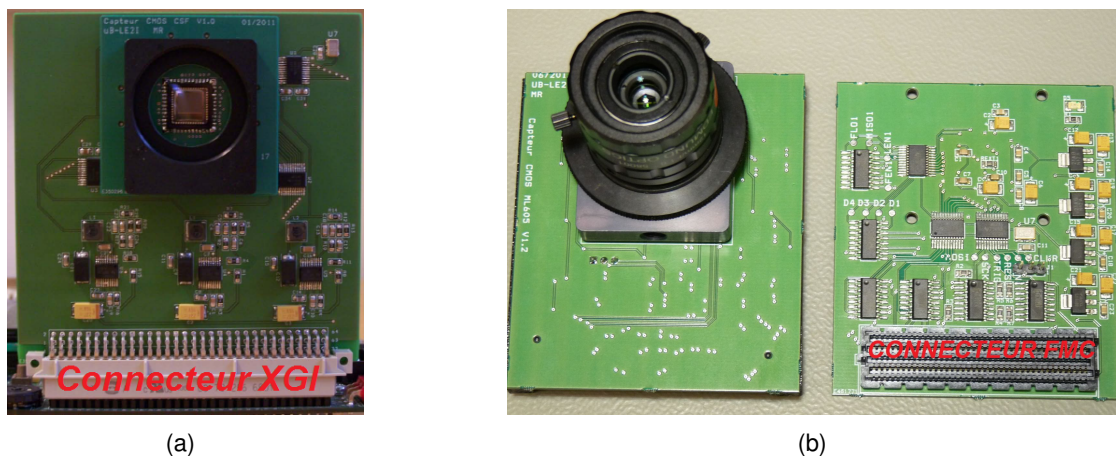
Méthode	Matériel	Capt.	Fus.	N	TMO	fps	Résolution	TR
<b>HDR</b>								
Akyüz [97]	GPU	✗	✓	9	✓	65	18 MPixels	✓
Kiser [91]	FPGA	✗	✗	-	✓	30	1,920 × 1,080	✓
Hassan [88]	FPGA	✗	✗	-	✓	60	1,024 × 768	✓
Mann [99]	FPGA	✓	✓	3	✓	120	1,280 × 720	✓
Kang [93]	Pentium 4	✓	✓	2	✓	15	1,024 × 768	✗
Tocci [82]	3 capteurs	✓	✓	3	✗	24/30	1,920 × 1080	✗
Unger [95]	Camera prog.	✓	✓	3	✗	25	512 × 512	✗
Ureña [96]	GPU/FPGA	✗	✗	-	✓	30/60	640 × 480	✓
Guthier [83]	CPU+GPU	✓	✗	-	✗	25	640 × 480	✓
Guthier [100]	GPU	✓	✓	-	✓	23	640 × 480	✓
Ali [85]	GPU	✓	✓	4	✗	398	640 × 480	✓
Castro [16]	Nokia N900	✓	✓	3	✗	25	848 × 480	✗
Ching-Te [92]	ARM SOC	✗	✗	3	✓	60	1,024 × 768	✓
Ke [98]	UMC 90nm	✗	✓	1	✓	60	720 × 480	✓
<b>Exp. Fusion</b>								
Bachoo [79]	CPU+GPU	✗	✓	3	-	20	1,600 × 1,200	✓
Gelfand [94]	Nokia N900	✓	✓	2-3	-	-	2,592 × 1,968	✗
Tico [80]	CPU	✗	✓	2-3	-	-	3,888 × 2,592	✗
Youm [81]	CPU	✓	✓	2	-	-	640 × 480	✗

**TABLE 3.2** – Tableau récapitulatif des systèmes matériels vidéo HDR ou HDR-like (ou Exposure Fusion) et des fonctionnalités qu'ils proposent et si ils sont ou non temps-réel (TR). La fonctionnalité de capture suppose la proposition d'une méthode originale permettant de sélectionner judicieusement les temps d'exposition.

bas, et une faible consommation ( $<200mW$ ). Cette faible consommation le rend adapté et utilisable pour des systèmes embarqués telles que les smart caméras, et les applications alimentées par batteries. Avec ses  $62dB$  de dynamique, il peut être considéré comme un capteur CMOS LDR standard. Notre architecture étant destinée à pouvoir s'adapter à un grand nombre de capteurs abordables présents sur le marché, le choix de l'utilisation de ce type de capteur conventionnel est assez judicieux.

En début de thèse, il a tout d'abord fallu construire une carte capteur, sous la forme d'une extension PCB sur mesure, qui puisse alimenter correctement le capteur et gérer ses bus d'entrées et de sorties. Ces cartes capteurs ont été réalisées grâce à l'aide précieuse de Matthieu Rossé, qui a conçu les "layouts" sous le logiciel Orcad, et aussi fait les commandes et réalisé les multiples tâches techniques. Ce début de thèse a donc été une période assez technique, dont le travail principal était d'étudier le fonctionnement du capteur via sa datasheet, de définir les principaux composants à incorporer sur la carte capteur, de réaliser la schématique d'une telle structure, etc. Deux types de carte capteur ont été construites au cours de la thèse, se différenciant notamment par leur différents connecteurs XGI ("Xilinx Generic Interface") et FMC ("FPGA Mezzanine Card"). Les connecteurs FMC sont plus polyvalents, puisqu'ils respectent une norme ANSI/VITA,

qui définit les modules de type mezzanine de connexion à un FPGA ou un autre appareil reconfigurable. Notre but a donc été de s'adapter aux différents types d'interfaçage présents dans le commerce. Deux photographies de ces cartes capteur sont présentes sur le Figure 3.10. Une description détaillée de la carte au connecteur XGI est montrée en Annexe F, avec la liste des composants, les schématiques et des photographies de la carte.



**FIGURE 3.10** – Les deux cartes capteur fabriquées au cours de la thèse. Elles supportent les versions de capteurs Sapphire - EV76C560 (1.3 MPixels) et Sapphire - EV76C570 (2 MPixels) de chez e2v. Les cartes sont déclinées en deux versions, une version adaptable pour les connecteurs XGI (a), et une version pour les connecteurs FMC (b).

### 3.8.2/ LE FPGA ET LE GPU : DES ARCHITECTURES PARALLÈLES

On remarque, d'après le Tableau récapitulatif 3.2, relatif à l'état de l'art des systèmes de vision HDR, que la plupart des systèmes temps réel actuels utilisent soit un GPU, soit un FPGA. Ces circuits ont une structure hautement parallèle, leur permettant d'accélérer grandement les traitements et qui les rendent efficaces notamment pour les traitements d'images et la vidéo. Les circuits programmables sont adaptés aux petites séries et au développement de prototypes tels que les nôtres.

Le GPU ("Graphics Processing Unit") est un circuit intégré sur carte graphique, en liaison directe avec une carte mère et un processeur. Ils ont été initialement conçus pour répondre au besoin d'accélération graphique, mais ont vite été employés pour le traitement parallèle. Ils sont en général peu coûteux en tant que tels, et sont utilisés comme coprocesseurs dans de nombreuses applications telles que les jeux vidéos, ou bien la simulation 3D de la physique, etc. Les GPUs offrent de bonnes performances, et sont facilement programmables dans un langage assez haut niveau comme le C, C++ (dans un environnement CUDA ou OpenCL). Lorsque l'application dédiée n'a pas d'interdépendance dans le flux de données, qu'il y a peu d'accès mémoire et que les calculs peuvent être effectués en parallèle, alors l'utilisation du GPU est judicieux. Les threads sont au-

tomatiquement gérés en interne sans forcément une intervention du programmeur. De même, celui-ci ne contrôle pas directement les processeurs parallèles, mais donne plutôt les directives au sein d'une API ("Application Programming Interface").

Les FPGAs sont de grands tableaux de cellules logiques, et sont assez flexibles dans le sens où le développeur peut facilement orienter son design vers plus ou moins de parallélisme. Il a accès à des données statistiques très précises sur son design, de telle sorte qu'il peut contrôler ses choix et réaliser des compromis entre les ressources utilisées et les performances désirées. Il a aussi accès directement aux ressources empruntées par l'application, et on peut même s'en servir dans l'optique de pouvoir concevoir un chip dédié de type ASIC ("Application Specific Integrated Circuit"), afin d'améliorer les performances, la fiabilité, mais surtout réduire les coûts dans une fabrication de composants à grande échelle. De plus, le développeur peut dupliquer autant de fois qu'il le veut son unité parallélisée, avec comme contrainte, la limite de surface exploitable imposée par la taille de son FPGA. Il peut également appliquer différents domaines d'horloge avec plusieurs fréquences dans tout son design. Néanmoins, même si les vendeurs de FPGAs comme Xilinx, mettent à disposition de nombreuses IPs ("Intellectual Properties") effectuant les tâches les plus communes, le développement d'un design de complexité moyenne est relativement coûteux. En effet, cette tâche est en général une mission qui demande un long travail et beaucoup d'heures dédiées à la manipulation du langage, de ses outils, et des techniques de design hardware.

Le GPU a un modèle de programmation informatique plutôt fixe. Le FPGA n'en a pas à proprement parlé, et la logique peut être personnalisée exactement pour les besoins de l'application. Les FPGAs requièrent généralement une certaine expérience dans la conception matérielle. Comparé au FPGA, le GPU peut être reconfiguré plus aisément, en réécrivant le code et en le recompilant dans la foulée, sans avoir besoin de connaissances préalables en hardware. Au niveau des interfaces, le GPU n'est interfaçable qu'avec le PCI Express (ou bien anciennement l'AGP et le PCI) pour le moment, alors que le FPGA possède de nombreux ports IOs ("In/Out"), compatibles en tension et en débits, ce qui le rend directement interfaçable avec n'importe quel chip externe (comme les codecs audio, le DVI, les mémoires externes, les capteurs divers, etc.).

Le FPGA peut, entre autres, effectuer des opérations pratiques de type bit à bit, chose que le GPU ne peut pas faire. Par exemple, une permutation de bit peut se faire directement sans aucun cycle sur FPGA. Du point de vue de la précision des calculs, les GPUs ont une importante capacité à gérer nativement les opérations en flottants IEEE-754 (simple et double précisions). Malheureusement, les FPGAs ne supportent pas nativement ce format, et utilisent plutôt les opérations arithmétiques en virgules fixes par défaut. Pour les développeurs voulant une précision supplémentaire, il est nécessaire d'implémenter les opérateurs en utilisant les ressources programmables intrinsèques au FPGA. Ces types de calculs consomment un volume de ressources qui sont à prendre en compte, et nécessitent une structure en pipeline (plus de 10 étages parfois), pouvant

créer une latence importante en sortie d'opérateur. Néanmoins, au vue de la technologie actuelle, rien n'empêche vraiment l'utilisation de l'arithmétique flottante au sein du FPGA, facilité par de nombreux IPs fournis par les vendeurs, et qui sont améliorés régulièrement. Une description de ces formats seront fait plus loin dans la thèse 4.5.2.1. Concernant les comparaisons entre les deux architectures, il semble peu évident de pouvoir juger objectivement des performances de l'une par rapport à l'autre ; toutefois, certaines études ont été mis au point à ce sujet [102].

Concernant la comparaison de prix entre FPGA et GPU, la question a soulevé de nombreux débats et de nombreuses controverses au sein de la communauté des développeurs hardware. Le gain en prix peut souvent aller dans le sens du choix du GPU ; mais à long terme, le prix augmente en raison d'une consommation en puissance plus élevée du GPU, ce qui tend à rendre le prix de revient du système plus élevé que celui du FPGA à long terme. Les GPUs sont souvent de gros appareils, difficilement intégrables dans un petit espace. L'utilisation de tel ou tel type de matériel est une question intéressante, soulevant une série de questions pouvant aider au choix final de la structure :

- Est-ce que l'application va être utilisée en tant que produit final, ou bien est-ce qu'elle va être incluse et interfacée avec d'autres IPs ?
- Est-ce qu'on envisage de faire migrer le design vers une future application embarquée sur ASIC ?
- Quels matériels et licences logicielles a-t-on à disposition ?
- Quel doit être le coût du système matériel de base ? Le coût de développement ?
- Quel est le niveau d'expérience du développeur FPGA et/ou GPU ?

De nos jours, d'autres points sont à prendre en considération, allant dans le sens de l'utilisation d'un FPGA plutôt que d'un GPU. Le développement sur FPGA est en train de basculer rapidement vers un développement software haut niveau de programmation en langage de type C (exemple des processeurs ARM embarqués sur Zynq [103], sur la plate-forme Zedboard [104]), ce qui peut réduire significativement les coûts de développement à terme. De plus, de nombreux outils performants de type "C-to-RTL", de passage automatique d'un langage haut niveau vers un langage bas niveau, sont en train de voir le jour et d'être perfectionnés.

Pour les raisons énoncées précédemment, nous préférons opter pour l'utilisation d'une architecture matérielle à base de FPGA.

### 3.8.3/ UN DISPOSITIF EXPÉRIMENTAL POUR LA VISION À GRANDE DYNAMIQUE

Pour mettre en place des algorithmes HDRs en temps réel, une caméra intelligente dédiée à base de FPGA a été conçue pour répondre aux besoins de calcul, de mémoire et de bande passante. Dans la suite de la thèse, cette architecture générique sera appelée HDR-ARtiSt ("High Dynamic Range-Advanced Real-time imaging System"), avec comme

paramètre  $P$ , le nombre d'expositions différentes utilisées. Le système HDR-ARtiSt sera testé pour une reconstruction à 2 et 3 images ( $P = 2$  et  $P = 3$ ).

La plate-forme HDR-ARtiSt est une caméra intelligente construite autour d'une carte capteur liée à une carte d'évaluation de chez Xilinx, équipé d'un FPGA Xilinx de type Virtex (voir Figure 3.11). La carte mère inclut une mémoire DDR SDRAM utilisée comme mémoire tampon pour les images capturées par le capteur. Plusieurs interfaces et périphériques standards sont également prévus pour relier le système au monde extérieur. Parmi ces interfaces, notre système de vision met en œuvre un contrôleur DVI afin d'afficher une vidéo sur un écran LCD, une liaison Ethernet, une liaison RS232, etc. Le contrôleur Ethernet sert à transmettre des trames UDPs ("User Datagram Protocol") vers un ordinateur hôte pour la récupération d'échantillons photographiques et vidéos.

Les FPGAs utilisés sont les Virtex 5 & Virtex 6. Ce sont de gros FPGAs, sélectionnés dans un premier temps pour s'affranchir des contraintes de ressources disponibles pour le prototypage. Les caractéristiques du Virtex 5 XC5VFX70T et du Virtex 6 XC6VLX240T sont définies dans le Tableau 3.3.

La série Virtex 5 FXT a la particularité d'avoir de hautes performances pour les systèmes embarqués, et d'héberger un cœur PowerPC 440 fonctionnant jusqu'à  $550MHz$ . Nous ne nous servons pas de ce processeur durant la thèse. Ce FPGA, dont le procédé de fabrication est le  $65nm$ , dispose de LUTs à 6 entrées. Chaque slice dispose de quatre LUTs et quatre registres d'un bit. Ses blocs RAM peuvent être configurés en double port. Des slices DSPs sont incorporés pour des opérations telles que les multiplications, additions, soustractions et accumulateurs.

Le Virtex 6, quant à lui, est fabriqué selon le processus  $40nm$ . Il possède des slices à quatre LUTs et huit registres. Les points qui le différencient du Virtex 5 ne résident essentiellement que dans les ressources ; les capacités du Virtex 6 sont globalement quatre fois supérieures à celles du Virtex 5, en termes de nombre de slices, LUTs, Block RAMs, DSPs, etc. Il peut embarquer un ou deux processeurs MicroBlaze. Le MicroBlaze est un softcore 32 bits d'architecture RISC, pouvant fonctionner jusqu'à  $200MHz$ , et est directement implémentable et fourni avec la licence Xilinx ISE ou Vivado.

Naturellement, les cartes d'évaluation ml507 et ml605 sont utilisées à des fins de prototypage rapide, et certains composants qui les composent ne sont nullement utilisés pour notre design. Il en résulte un prototype caméra exagérément dense et assez volumineux ; mais le système pourrait bien évidemment être miniaturisé par la suite.

### 3.9/ CONCLUSION

Au terme de ce chapitre, nous avons sélectionné l'architecture matérielle que nous utiliserons pour contenir notre futur système HDR-ARtiSt. Concernant nos contraintes ma-

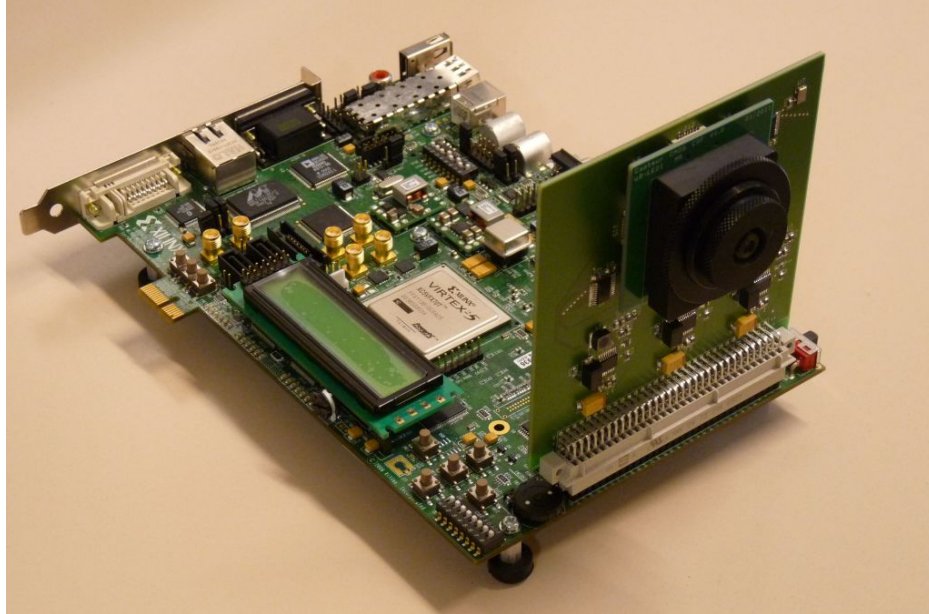


	<i>Virtex 5 (sur ml507)</i> <i>XC5VFX70T</i>	<i>Virtex 6 (sur ml605)</i> <i>XC6VLX240T</i>
<b>Année</b>	2008	2009
<b>Taille</b>	35 × 35mm	35 × 35mm
<b>Nombre de Slices</b>	11200	37680
<b>Nombre de Block RAM 18Kb</b>	296	832
<b>Nombre de Block RAM 36Kb</b>	148	416
<b>Nombre de slices DSP48E</b>	128	768
<b>Nombre d'I/Os</b>	640	720
<b>Processeur embarqué</b>	PowerPC (en dur)	MicroBlaze
<b>Fréquence Maximum (Clocks, DSP, bloc mémoire)</b>	550MHz	600MHz
<b>RocketIO GTX jusqu'à 6.5Gb/s</b>	16	24

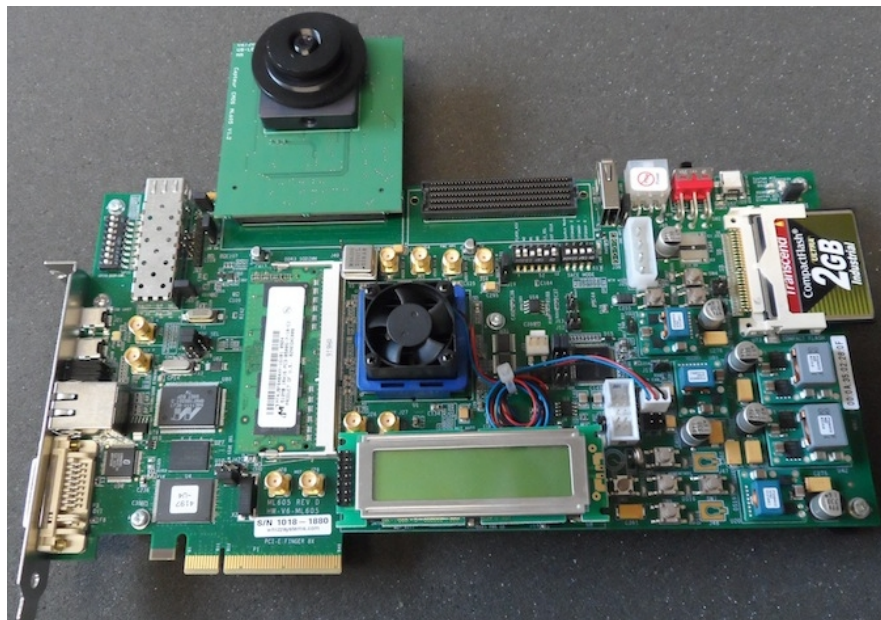
TABLE 3.3 – Résumé des particularités des deux FPGAs utilisés au cours de la thèse.

térielles et techniques, nous avons conclu sur l'utilisation d'une architecture à base de FPGA plutôt qu'un système basé sur GPU. De plus, le choix a été fait de réaliser les travaux en utilisant uniquement le langage VHDL et Verilog.

Reste maintenant à sélectionner judicieusement les algorithmes HDRs, et surtout à trouver des solutions pour outrepasser les contraintes techniques de capture, de timing pour le temps réel, de stockage des pixels, de bande passante, etc. Nous définirons donc les méthodes et les outils empruntés par notre système de vision HDR dans le chapitre suivant.



(a)



(b)

**FIGURE 3.11** – HDR-ARTiSt est prototypé sur des carte d'évaluation embarquant un FPGA de type Virtex de chez Xilinx. Les cartes capteur ont été fabriquées par nos soins. Elles accueillent un capteur CMOS EV76C560 1.3 Mégapixels de la société e2v. Les deux différentes "cartes mères" ici sont la ml507 (a) et la ml605 (b).





## HDR-ARTIST : UNE CAMÉRA HDR

### 4.1/ INTRODUCTION

Le monde de l'imagerie HDR se déplace doucement vers une direction où la mise en œuvre est temps réel, où une image HDR peut être produite immédiatement après la capture du set d'images LDRs. Ce chapitre décrit l'architecture complète d'une caméra intelligente que nous avons élaborée : *HDR-ARtiSt* ("High Dynamic Range - Advanced Real-time imaging System"). Elle produit un flux vidéo HDR en temps réel à partir de plusieurs images acquises avec des temps d'exposition différents. Un système de capture d'images intelligent a été défini. Une unité de gestion de mémoire spécifique permet d'assister et d'alimenter le processus HDR en flux vidéos bruts non compressés, et d'améliorer la fluidité finale de la vidéo HDR. Comme décrit dans la Section 3.8, cette caméra intelligente est construite autour d'un capteur d'image CMOS standard de chez e2v, et d'un FPGA de chez Xilinx. Les deux designs implémentés sur les cartes ml507 et ml605 sont similaires, seuls les composants connectés au FPGA et inclus avec les cartes de développement changent. Nous présenterons le système HDR-ARtiSt complet, qui est commun et implémentable indifféremment sur les deux plates-formes matérielles. La contribution principale de ce travail est la génération d'une nouvelle architecture FPGA embarquée qui produit un flux vidéo HDR en temps réel en noir et blanc, mais aussi en couleur, avec une résolution de  $1280 \times 1024$  pixels à 60 ips. Un contrôleur DVI décrit en VHDL a été également codé afin de pouvoir afficher les vidéos en direct sur un moniteur LCD et visualiser nos résultats. La caméra ainsi conçue pourra éventuellement intégrer d'autres modules algorithmiques plus complexes, de type traitement d'image, ou pourra même être connectée à un PC standard afin de gérer les flux vidéos en sortie. Notre caméra embarque plusieurs fonctionnalités :

- La capture
- La multiple auto exposition (MAE)
- La création d'images HDR
- La compression de dynamique
- L'affichage des données
- Le transfert de données vers un PC fixe via Ethernet.

Ce travail est une contribution originale par rapport à l'état actuel de la technique. L'architecture proposée permet d'obtenir entre autres un flux vidéo HDR directement visible et sans latence apparente pour le spectateur, le tout à une grande résolution et une cadence de  $60\text{ips}$ . La résolution et la cadence ne sont pas limitées à celle de notre capteur, et le système HDR-ARTiSt peut être adapté à d'autres capteurs disposant d'une résolution différente.

Le chapitre est organisé comme suit : la Section 4.2 mettra en lumière l'architecture modulaire globale du système. Dans la Section 4.3, nous présenterons notre méthode de multiple auto exposition (MAE), basée sur la capture intelligente d'images avec des temps d'exposition configurés directement en fonction de la dynamique de la scène. La Section 4.4 décrira notre gestion spécifique de la mémoire, avec l'unité de gestion complète conçue afin d'administrer la limitation de la bande passante de la mémoire, et d'alimenter le module HDR en flux de vidéos parallèles. Une description et une évaluation des algorithmes utilisés se fera au travers d'un logiciel réalisé par nos soins, *HDR-Lab*. Les simplifications des algorithmes et leur mise en œuvre seront détaillées dans la Section 4.5. Nous terminerons par une conclusion dans la Section 4.6, au terme de laquelle nous aurons défini l'ensemble des blocs de l'architecture. Les implémentations à deux images et trois images ( $P = 2$  et  $P = 3$ ), les résultats et les comparaisons, seront présentés dans le Chapitre 5 qui suivra.

## 4.2/ SYSTÈME GLOBAL

La Figure 4.1 représente notre architecture globale nécessaire à l'élaboration du système vidéo. Il est composé d'un capteur CMOS qui envoie les images au FPGA. L'auto exposition multiple se charge de contrôler les registres du capteur afin qu'il envoie séquentiellement les images LDRs avec des temps d'exposition appropriés à la capture de la dynamique de la scène. Celui-ci devra fonctionner en parallèle à la réception et au traitement d'image envisagé. De plus, un tel système doit avoir un module de contrôle d'exposition qui gère les registres du capteur très rapidement, afin que le calcul en temps réel des images HDRs ne soit pas biaisé par des images mal sélectionnées. Le module MMU est un système de management de la mémoire conçu pour assister le module HDR afin de l'alimenter en flux d'images LDRs. La double flèche allant à la fois à la mémoire et au module HDR, signifie que les flux d'images proviennent, à chaque instant, du capteur et de la mémoire. La mémoire agit en quelque sorte à la manière de capteurs synchronisés avec le capteur CMOS, en fournissant en parallèle les images LDRs enregistrées précédemment. La MMU est assistée d'un certain nombre de BRAMs ("Block RAMs"), servant à faire correspondre les domaines de fréquences et bufferiser les pixels ligne par ligne. On remarque que le nombre d'images entièrement stockées est de  $P - 1$ . Le bus de communication avec la mémoire est d'une taille de 64 bits. Nous implémenterons le système sur une mémoire DDR2 SDRAM, et sur une mémoire DDR3 SDRAM. Ces

deux types de mémoires seront utilisées respectivement pour les plates-formes ml507 et ml605. Nous détaillerons le fonctionnement de l'interface mémoire dans la Section 4.4, appuyés de l'Annexe A. L'interface Ethernet nous servira lors de la réalisation d'une interface sous Qt nommée *HDR-ViewFinder*, que nous décrirons plus loin dans la thèse, dans la Section 5.9.2 du Chapitre 5.

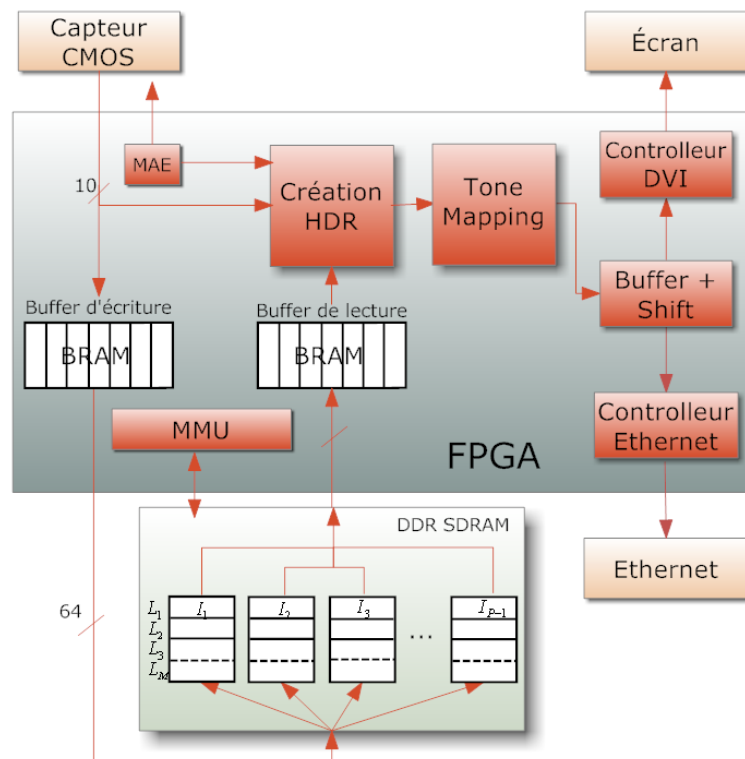


FIGURE 4.1 – Vue globale du système HDR-ARTiSt.

### 4.3/ AUTO EXPOSITION MULTIPLE (MAE)

Le mode bracketing de capture automatique d'expositions présent dans les appareils photo numériques actuels, est une technique répandue dans le domaine de la photographie amateur. Le but est de pouvoir sélectionner automatiquement et à des intervalles réguliers, des valeurs expositions multiples autour d'une exposition dite optimale. Notre contrôle d'exposition multiple (MAE) est légèrement différent parce que notre objectif est plutôt d'optimiser le nombre de capture, en essayant d'avoir le maximum de pixels bien exposés ; plutôt que de fixer les temps d'exposition à des intervalles réguliers (tout les  $2EV$  par exemple). Notre travail est relativement proche du contrôle d'exposition de Kang [93] et du système de double exposition initié par Alston et al. [105]. Dans notre cas, les paramètres d'exposition sont continuellement adaptés aux changements de luminosité de la scène, puisque les temps d'exposition peuvent être modifiés directement après la réception d'une nouvelle image issue du capteur. L'estimation des meilleurs temps

d'exposition est faite à partir des statistiques de la scène fournies par notre capteur dans le flux de données (histogramme codé en queue de trame). Notre capteur est capable de nous envoyer l'histogramme de l'image complète. L'histogramme de chaque image nous permet d'avoir un aperçu en temps réel de l'étendue totale de luminosité qui est enregistrée.

Lorsque l'assemblage HDR est effectué après une auto exposition ordinaire, les algorithmes traditionnels échouent par manque de dynamique capturée. À la manière de l'œil humain, notre caméra doit chercher à s'adapter automatiquement en fonction de l'éclairage ambiant. Nous présenterons une approche similaire à un algorithme issu de l'état de l'art, adapté à nos besoins en termes de matériel et de temps réel.

Gelfand et al. [94] utilisent une API et une FCAM programmable afin de capturer deux images, en alternant une exposition courte et longue. Lorsque les deux expositions sont stables, l'ajustement d'exposition est terminé et ils peuvent alors effectuer des calculs de traitement HDR. Ils ne tiennent pas compte des changements futurs des conditions lumineuses susceptibles d'intervenir, ce qui pourrait corrompre la valeur de la dynamique de luminances visualisée. De la même manière, nous nous assurons que moins de 10% des pixels soient saturés dans la partie haute de la plage tonale du capteur pour l'image d'exposition courte. Si un trop grand nombre de pixels est saturé haut, le temps d'exposition est réduit via les registres du capteur pour satisfaire nos besoins à la prochaine intégration. De même, nous exigeons que moins de 10% des pixels soient saturés bas concernant la longue exposition. Si trop de pixels sont sombres, la durée d'exposition est augmentée. Nous ne prenons pas comme valeur 0% car la stabilisation des expositions ne pourraient pas se faire dans certaines conditions. En effet, nous avons repéré qu'une contrainte trop stricte peut entraîner une impossibilité au capteur de trouver un régime stable, ce qui conduit à des effets indésirables de type flicker (ou scintillements). Nous préférons donc garder un seuil de tolérance de 10% pour ne pas avoir de dysfonctionnement visible.

Notre capteur ayant une sortie de pixels codée sur 10bits, nous fixons une plage tonale pour laquelle nous considérons les pixels comme saturés si :

- $Z \in \{0, 64\} \rightarrow$  Pixel saturé en niveau bas
- $Z \in \{959, 1023\} \rightarrow$  Pixel saturé en niveau haut.

Gelfand et al. arrêtent ce processus itératif lorsque les deux expositions sont stables, et utilisent ces valeurs pour la capture de l'intégralité de leur photographies. Une telle approche est optimale pour la capture d'une seule scène fixe, mais cette façon de procéder ne peut pas convenir dans un contexte vidéo. Ainsi, dans notre approche, nous décidons de continuellement mettre à jour l'ensemble des temps d'exposition de chaque trame afin de gérer quasi instantanément tout changement des conditions d'éclairage.

Considérons le cas pour  $P = 2$ , où l'image  $I_L$  a pour durée d'exposition  $\Delta t_L$ , et  $I_H$  a pour durée d'exposition  $\Delta t_H$ . La Figure 4.2 illustre la façon dont l'algorithme MAE estime automatiquement la nouvelle durée d'exposition pour l'exposition  $I_L$ , après variation brutale

de l'éclairage dans la scène. L'image est volontairement réduite à une dynamique de 8 bits par pixel pour pouvoir afficher l'exemple (on supprime les 2 bits de poids faibles). Lorsque la lampe s'allume, le nombre de pixels saturés blanc augmente de façon significative (15,3% pour la Figure 4.2(a)<sup>1</sup>). Ainsi, la MAE diminue le temps d'exposition pour obtenir moins de 10% de pixels saturés dans l'image  $I_L$ . Dans notre cas réel, on obtient au final une durée  $\Delta t_L$  stable pour laquelle 5,4% des pixels sont saturés blanc, comme illustré dans la Figure 4.2(b). Une approche similaire est utilisée pour estimer le temps d'exposition haut  $\Delta t_H$ , voir la Figure 4.3. Les histogrammes correspondant aux images sont visibles sur les Figures 4.2(c) et 4.2(d) pour  $I_L$  et 4.3(c) et 4.3(d) pour  $I_H$ . Du fait de la présence d'un trop grand nombre de pixels noirs dans les images, nous présentons volontairement les histogrammes en rapport avec les pixels à l'intérieur des cadres roses, ceci pour avoir une représentation statistique assez visible.

Afin d'obtenir un temps de convergence correcte, les temps d'exposition sont automatiquement ajustés en utilisant deux niveaux de seuils différents : une petite plage de variation pour les petits changements d'illumination, et une plus grande plage de variations pour les grands changements d'illumination. Pour le cas particulier où  $P = 3$ , le temps d'exposition intermédiaire ou moyen  $\Delta t_M$ , est calculé à partir d'une combinaison des deux autres. On utilise la relation suivante :

$$\Delta t_M = \sqrt{\Delta t_L \times \Delta t_H}. \quad (4.1)$$

Les formules qui calculent la valeur d'exposition  $EV$  et le temps d'exposition sont les suivantes :

$$EV = \log_2 \frac{\text{f-number}^2}{\Delta t}, \quad (4.2)$$

$$\Delta t = \frac{\text{f-number}^2}{2^{EV}}, \quad (4.3)$$

où f-number est l'ouverture.

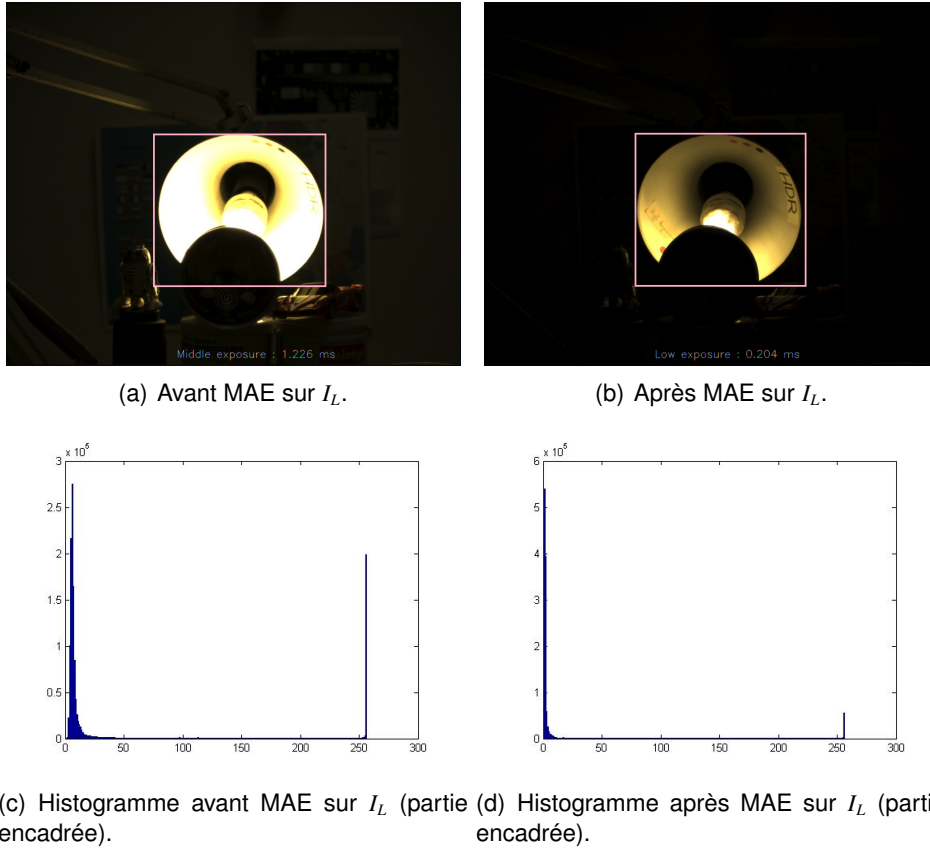
Pour démontrer la relation 4.1, on prend f-number = 1, et on a :

$$\Delta t_M = \sqrt{\frac{1}{2^{EV_H} \times 2^{EV_L}}} = \sqrt{\frac{1}{2^{EV_H+EV_L}}} = \frac{1}{2^{\frac{EV_H+EV_L}{2}}} = \frac{1}{2^{EV_M}}, \quad (4.4)$$

où  $EV_M$  est l'exposition moyenne théorique. Prenons par exemple  $EV_H = 2$  et  $EV_L = 10$ , on a  $\Delta t_H = 0.25s$  et  $\Delta t_L = 0.001s$ . La valeur moyenne théorique  $\Delta t_M$  pour laquelle  $EV_M = \frac{EV_H+EV_L}{2} = 6$  est  $\Delta t_L = 0.167s$ .

Comme nous travaillons avec une architecture hautement parallélisable, nous pouvons faire d'autres estimations de la plage dynamique de la scène pendant le traitement HDR. Si nécessaire, nous corrigeons les temps d'exposition lorsque les conditions d'éclairage changent. Nous aborderons en détail et d'une manière un peu plus formelle notre algo-

1. Les images sont issues de la caméra HDR-ARtiSt.



**FIGURE 4.2** – La Figure (a) est composée 15,3% de pixels saturés blanc lorsque la lampe est allumée. Après l'opération d'auto exposition, la Figure (b) présente 5,4% de pixels saturés blancs.

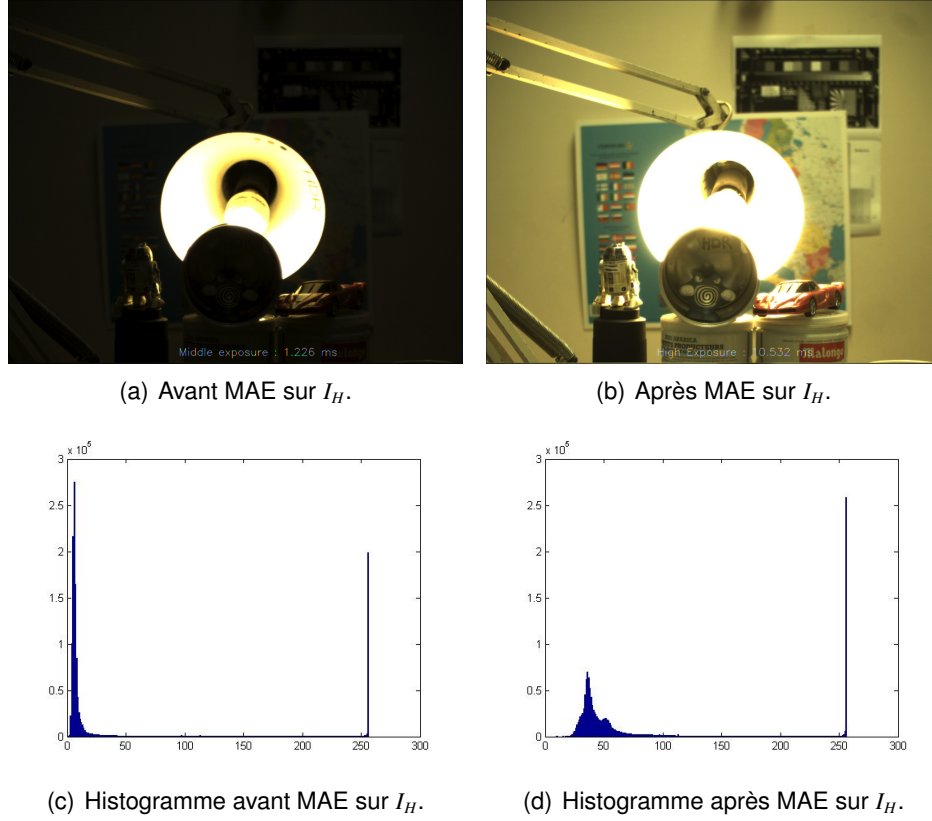
rithme d'auto exposition dans la Section 4.3.1 suivante.

#### 4.3.1/ ALGORITHMIE POUR LA MAE

Les histogrammes que nous fournit le capteur sont codés en 64 catégories, avec une largeur de 16bits par catégorie, comme le montre la Figure 4.4. Selon l'image que l'on reçoit, une exposition faible  $I_L$  ou une exposition forte  $I_H$ , nous appliquons ces fonctions :

$$Q_L = \sum_{h=1}^{h=4} \frac{q(h)}{N} \quad Q_H = \sum_{h=60}^{h=64} \frac{q(h)}{N}, \quad (4.5)$$

où  $Q_L$  et  $Q_H$  sont respectivement la proportion de pixels présents sur une partie spécifique de l'histogramme, parmi les  $N$  pixels que composent une image.  $q_h$  est la quantité de pixels dans chaque catégorie  $h$  de l'histogramme. Le calcul est fait avec les quatre premières et les quatre dernières catégories dans les images  $I_H$  et  $I_L$ . Les pixels de sortie étant codés sur 10-bit (entre 0 et 1023), quatre catégories correspondent à un intervalle de 64 valeurs de pixels. Ensuite, nous calculons deux paramètres pour les deux temps



**FIGURE 4.3** – La Figure (a) est composé 38,3% de pixels saturés noir lorsque la lampe est allumée. Après l'opération d'auto exposition multiple, la Figure (b) présente 0,2% de pixels saturés dans le noir.

d'exposition extrêmes comme ceci :

$$\delta Q_{L/H} = |Q_{L/H} - Q_{L/H,req}|, \quad (4.6)$$

où  $Q_{L/H,req}$  est la quantité nécessaire de pixel pour la partie de l'histogramme concernée (10% de  $N$ ).  $\delta Q_{L/H}$  évalue l'écart entre la quantité de pixels courant et la quantité souhaitée. Une fois que nous avons ces paramètres, on peut prendre une série de décisions pour la capture des images à  $t + 1$  :

$$\Delta t_{L/M/H,t+1} \leftarrow MEC(\Delta t_{L/H,t}), \quad (4.7)$$

$$\Delta t_{L,t+1} = \begin{cases} \Delta t_{L,t} \pm 1x & \text{pour } \delta Q_L > thr_{Lm} \\ \Delta t_{L,t} \pm 10x & \text{pour } \delta Q_L > thr_{Lp} \end{cases} \quad (4.8)$$

$$\Delta t_{H,t+1} = \begin{cases} \Delta t_{H,t} \pm 1x & \text{pour } \delta Q_H > thr_{Hm} \\ \Delta t_{H,t} \pm 10x & \text{pour } \delta Q_H > thr_{Hp} \end{cases} \quad (4.9)$$

$$\Delta t_{M,t+1} = \sqrt{\Delta t_{L,t} \Delta t_{H,t}}, \quad (4.10)$$



où  $\Delta t_{L/M/H,I}$  sont les valeurs de temps d'exposition des images courantes  $I_L$ ,  $I_M$  et  $I_H$ .  $thr_m$  (minus) et  $thr_p$  (plus) sont les deux valeurs de seuils qui correspondent aux deux niveaux différents d'action sur l'ajustement des temps d'exposition. Ces seuils vont directement influencer sur la vitesse de transition vers un état stable des valeurs d'exposition. Par exemple, lorsque l'on a un grand bon positif dans la dynamique de la scène en un espace de temps court, on dépassera plutôt le seuil  $thr_p$ , et on affectera les registres du capteur en conséquence.  $x$  est un temps correspondant au temps nécessaire à l'intégration d'une ligne du capteur. En l'occurrence, les temps d'intégration de notre capteur se configurent en nombre de lignes. Dans notre cas,  $x = 15.72\mu s$ .

$\Delta t_{L/M/H,I+1}$  sont les durées d'exposition futures qui seront programmées dans le capteur pour toutes les trames suivantes  $I_{L/M/H}$ . Le capteur dispose jusqu'à quatre ROIs configurables. Ces ROIs sont envoyés séquentiellement durant la capture ( $ROI_1 \rightarrow ROI_2 \rightarrow ROI_3 \rightarrow ROI_4 \rightarrow ROI_1$ , etc.). Nous n'utilisons que trois ROIs au maximum. Il nous est aisé de savoir quelle est l'image basse, moyenne et haute que l'on reçoit du capteur, car chaque image est envoyée avec son numéro de ROI fournie dans l'entête de trame. Un morceau de l'entête composé de 2 bits est codé tel que : "00"  $\rightarrow ROI_1$ , "01"  $\rightarrow ROI_2$  et "10"  $\rightarrow ROI_3$ , chaque ROI correspondant bien évidemment aux trois images pleine résolution d'expositions différentes.

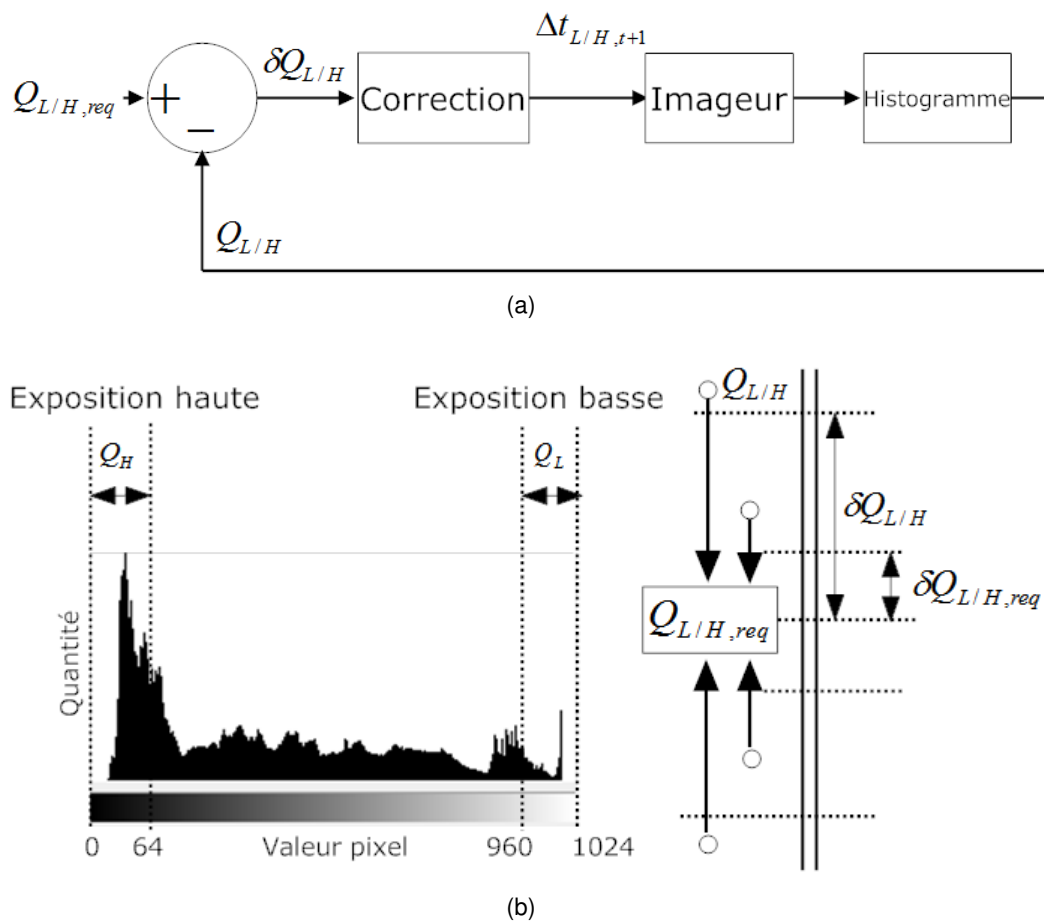
Une fois que nous avons les deux expositions extrêmes  $\Delta t_H$  et  $\Delta t_L$  pour lesquelles la condition des 10% de pixels saturés en noir ou en blanc est respectée, on peut estimer la dynamique de la scène capturée par l'équation suivante :

$$D_y = \frac{\Delta t_H}{\Delta t_L}, \quad (4.11)$$

où  $D_y$  est la dynamique calculée. Elle est utilisée comme un indicateur de la dynamique de la scène observée.

Nous montrerons les résultats des différents flux vidéos parallélisés, traités par notre algorithme MAE, au travers de la Section 5, et d'aperçus en temps réel (voir site internet fourni en annexe de la thèse). La configuration des registres du capteur se fait à l'aide du protocole SPI ("Serial Peripheral Interface bus"). L'interface est composée de 4 fils, SLK (horloge), MOSI (données vers le capteur), MISO (données vers le FPGA) et CSN ("Chip Select"). Nous ne détaillerons pas le fonctionnement du bus.

De plus, une interface de plus haut niveau a été mise en place. L'utilisateur peut basculer entre les expositions faible, moyenne, haute, ou directement la vidéo HDR après tone mapping via notre interface Ethernet *HDR-ViewFinder* (voir Figure 5.6). Nous pouvons alors nous assurer que les expositions appropriées ont été correctement sélectionnées par le MAE, de telle sorte que tous les détails dans les zones sombres et lumineuses de la scène sont bien représentés, et que le module MAE fonctionne bien. Nous pouvons aussi récupérer les images brutes via cette interface, pour y effectuer des tests et calculer les histogrammes sous Matlab à titre de vérification.



**FIGURE 4.4** – Système de contrôle de l'exposition MAE. Les informations et les histogrammes sont envoyés par le capteur. L'histogramme est stocké et utilisé pour prendre une décision sur chaque temps d'exposition. Pour l'exposition longue  $I_H$ , le seuil faible  $Q_{H,req}$  est défini dans le but de s'assurer que moins de 10% de la totalité des pixels ont des valeurs inférieures à 64. Sinon,  $\Delta t_{H,l+1}$  sera mis à jour afin de réduire le temps d'exposition.

Une fois l'auto exposition faite, il nous faut recueillir et gérer les multiples images que nous envoie le capteur. Pour cela, une unité de management de la mémoire (MMU) a été conçue. C'est ce que nous allons voir dans la prochaine partie.

## 4.4/ MANAGEMENT DE LA MÉMOIRE DÉDIÉ HDR-MMU

La technique envisagée de création de vidéo HDR a besoin de quatre étapes successives :

1. La capture de  $P$  images uniques sélectionnées par la MAE
2. Le stockage de  $P - 1$  images en mémoire
3. Le calcul de la carte des radiances pendant l'acquisition de la  $P^{\text{ième}}$  image et la lecture des  $P - 1$  images depuis la mémoire.
4. L'exécution de l'opération de tone mapping directement pendant la réception de la  $P^{\text{ième}}$  image.

Le principal inconvénient de la génération HDR par la multiple exposition est le débit de sortie limité. Comme illustré sur la Figure 4.5, avec  $P = 3$  images uniques (basse, moyenne et haute expositions) capturées à 60 ips, la sortie d'une caméra HDR standard est considérablement limitée à 20 images par seconde : la première image HDR  $H_1$  est calculée à partir des cartes de radiances  $I_1$ ,  $I_2$  et  $I_3$ , la deuxième  $H_2$  à partir de  $I_4$ ,  $I_5$  et  $I_6$ , etc.

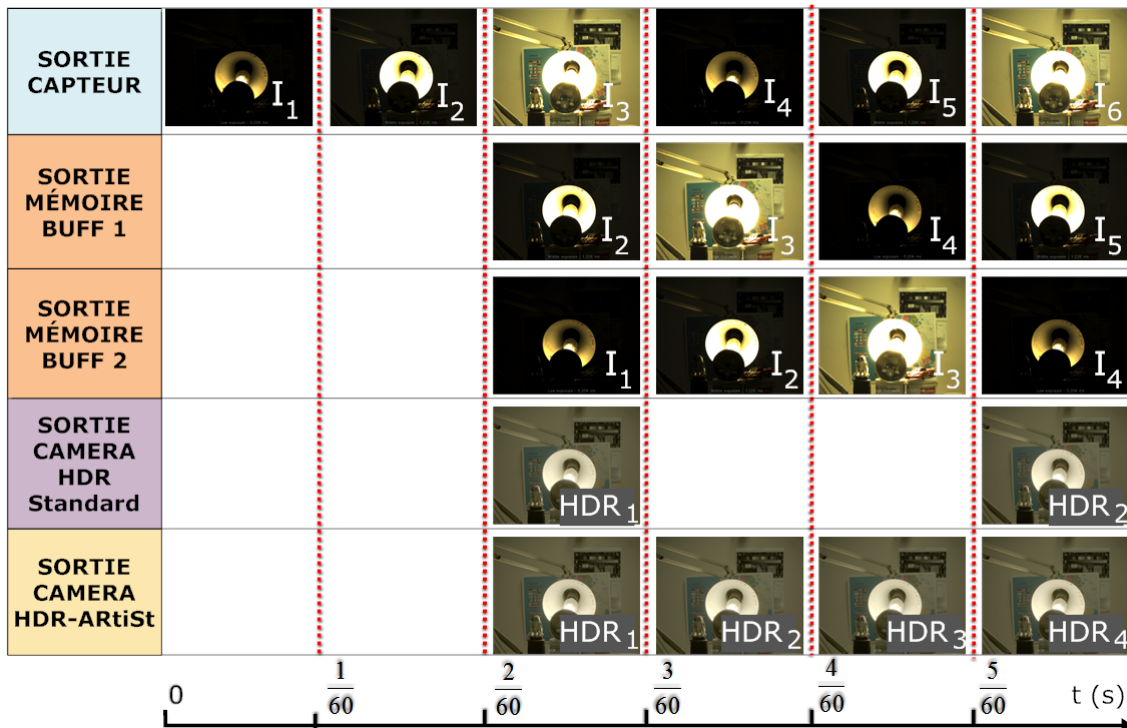
Pour contourner cette limite, nous proposons une gestion de mémoire spécifique : le *HDR-MMU*, dans la mesure où l'on crée une nouvelle image HDR à la cadence du capteur et en temps réel, en mettant à jour à chaque instant l'information "fraîche" issue du capteur. La méthode peut être apparentée à la technique de "memory swapping" [106].

Nous définirons dans un premier temps la notion de vidéo en temps réel.

### 4.4.1/ DÉFINITION D'UNE VIDÉO TEMPS RÉEL

Un système temps réel est construit dans le but de pouvoir gérer le processus qui le compose à une vitesse contrôlée, dans un intervalle de temps constant, intervalle qui puisse contribuer à la bonne évolution du processus globalement maîtrisé.

Nous définirons ici le terme de "vidéo temps réel". La vidéo est dite temps réel, si la latence entre l'entrée d'un pixel dans le système, et la sortie de ce même pixel après traitement est fixe et petite. Par latence petite, nous sous-entendons ici une durée pour laquelle le spectateur ne perçoit pas le temps de réponse du système. Autrement dit, le processus de traitement d'une vidéo ne met pas plus de temps à se réaliser par rapport à la longueur de la vidéo en elle-même. Par exemple, si une vidéo dure 40 min, le traitement d'image appliqué à cette vidéo ne devra pas durer plus de 40 min.



**FIGURE 4.5** – Comparaison entre les framerates de la technique HDR standard et de notre technique. Un pipeline d'images est mis en place afin de délivrer à chaque instant plusieurs flux vidéos en parallèle.

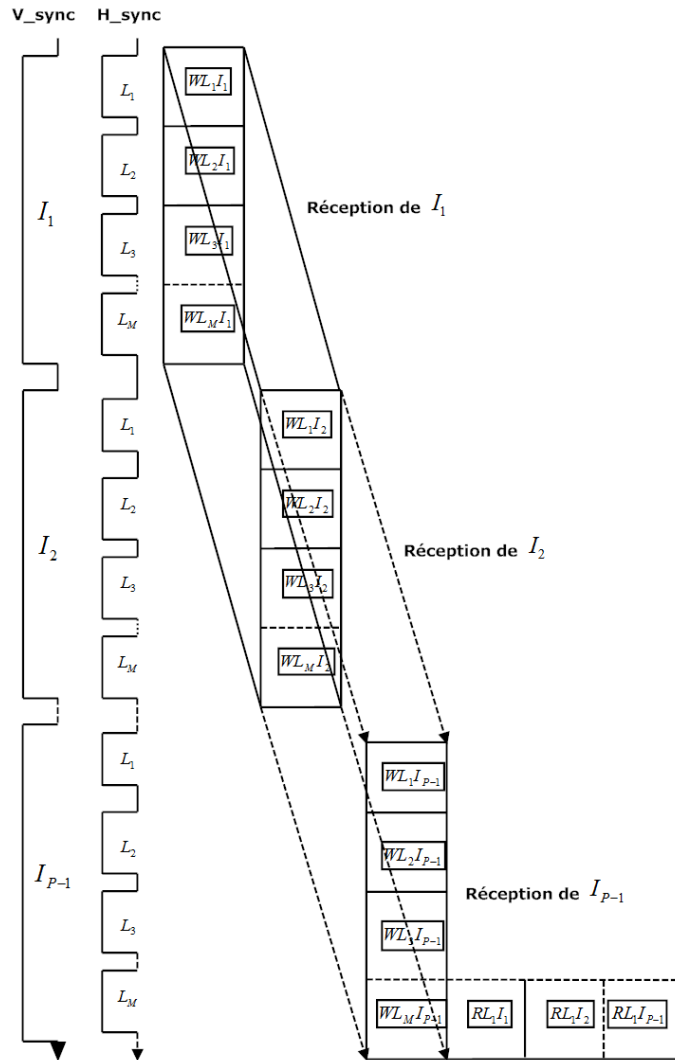
Dans notre cas, le traitement vidéo sur chaque pixel se fait à la cadence de notre capteur, soit à une fréquence d'environ 110 MHz, avec une latence de quelques microsecondes par rapport à l'entrée. Si l'on fait l'expérience en présentant deux moniteurs à un spectateur, l'un affichant la vidéo originale du capteur, et l'autre affichant la vidéo traitée, la latence serait imperceptible à l'œil nu.

Dans le domaine de la vidéo, environ 16 ips sont le strict minimum pour avoir l'illusion de fluidité dans une scène où la dynamique des mouvements locaux n'est pas trop importante.

Nous détaillerons dans la prochaine partie comment les images sont stockées et rendues au système dans le but d'obtenir de multiple flux vidéos en parallèle.

#### 4.4.2/ PRINCIPE

Comme on peut le voir sur la Figure 4.5, l'image  $HDR_2$  est construite à partir de  $I_2$ ,  $I_3$  et  $I_4$ ;  $HDR_3$  à partir de  $I_3$ ,  $I_4$  et  $I_5$ , etc. La gestion de la mémoire est appelée MMU, pour "Memory Management Unit", avec  $P$  le nombre d'images. Elle gère en permanence le stockage des  $P - 1$  images, la plus ancienne image étant systématiquement écrasée dans la mémoire par la nouvelle image acquise. Dans un même temps, la MMU gère la lecture de ces  $P - 1$  images en sortie de capteur afin d'alimenter le processus de création

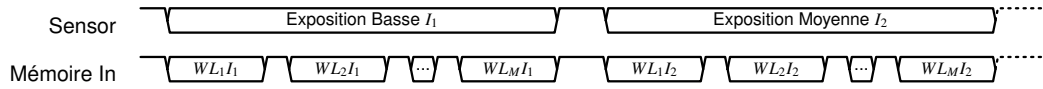


**FIGURE 4.6** – Initialisation de la MMU, avec le stockage des  $P - 1$  images dans la mémoire et la lecture des  $P - 1$  premières lignes des  $P - 1$  premières images.

## HDR.

A cet effet, une stratégie de partage du temps est utilisée pour stocker et lire les différents flux vidéos dans la mémoire SDRAM. Des mémoires tampons de type BlockRAMs (BRAMs) sont embarquées par le FPGA, et sont nécessaires pour stocker les lignes de pixels temporairement, et gérer le partage du bus de données de la mémoire SDRAM. Pour relever le défi de contrainte temps réel et réduire au minimum la taille des mémoires tampons, la MMU effectue les lectures/écritures ligne par ligne, pour transférer les différents flux.

Avant de pouvoir envisager d'effectuer le calcul de la première image HDR, l'unité de mémoire nécessite d'être mise en place. En effet, lorsque le système de vision vient d'être activé, une étape d'initialisation préalable a pour but de capturer et stocker  $P - 1$  images dans la mémoire SDRAM. La Figure 4.6 montre l'initialisation de la MMU. Les



**FIGURE 4.7** – Initialisation du management de la mémoire. Le capteur envoie séquentiellement une exposition basse ( $I_L$ ) et moyenne ( $I_M$ ). On peut noter l'écriture dans la mémoire des  $M$  lignes, pour les deux premières images.

synchronisations  $h_{sync}$  et  $v_{sync}$  sont actives à l'état bas. Avec  $0 < m \leq M$  et  $0 < p < P$ , on a :

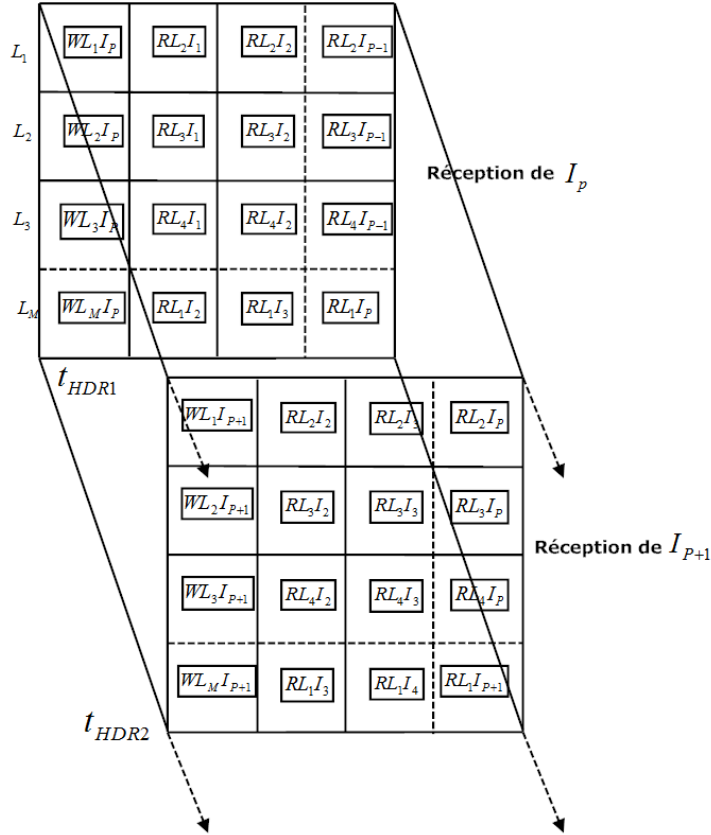
- $WL_M I_1$  : écriture des  $M$  lignes de l'image  $I_1$
- $WL_M I_2$  : écriture des  $M$  lignes de l'image  $I_2$
- ...
- $WL_M I_{P-1}$  : écriture des  $M$  lignes de l'image  $I_{P-1}$

Après avoir reçu la dernière ligne  $L_M$  de l'image  $I_{P-1}$ , les premières lignes  $L_1$  de chaque trame mémorisée précédemment ( $RL_1 I_p$  avec  $0 < p < P - 1$ ), sont lues pendant la synchronisation verticale des images et bufferisées dans les BRAMs, afin d'être rapidement disponibles pour le processus de création de la première image  $HDR_1$  (Voir Figure 4.5). Une mise à plat du schéma de la Figure 4.6 est montrée au travers de la Figure 4.7.

Du fait de la conception des BRAMs, il est nécessaire de combiner 2 BRAMs pour avoir une sortie sur le nombre de bits que l'on désire, de telle manière à utiliser au maximum la largeur du bus de la mémoire. La description de la MMU pour  $P = 3$  est présente dans l'Annexe A.

La seconde étape est gérée par le noyau MMU et commence au début de la capture de  $I_P$ . Entre chaque intervalle de ligne (synchronisation horizontale du capteur), la ligne courante  $L_m$  est écrite dans la mémoire SDRAM, alors que toutes les lignes  $L_{m+1}$  des  $P - 1$  images précédentes sont lues et stockées dans les BRAMs, tel que représenté sur la Figure 4.8. Avec une telle technique, lorsque le capteur nous envoie une nouvelle ligne de pixels, le processus HDR a un accès simple et efficace à la ligne courante et aux lignes des trames précédentes par l'intermédiaire des mémoires tampons. Puis, la première image  $HDR_1$  est obtenue à la fin de la capture de l'image  $I_P$  à  $t = t_{HDR_1}$  (voir Figure 4.8). Le processus est ensuite répété avec l'image  $I_{P+1}$  et le calcul de la deuxième trame HDR à partir de  $I_2, I_3, \dots, I_{P+1}$  à  $t = t_{HDR_2}$ , etc. La Figure 4.9 est une autre représentation de l'interface mémoire développée. Elle met en plus en évidence la sortie HDR affectée d'une latence. Cette latence provient du traitement algorithmique des pixels, et sera quantifiée lors des Sections 5.6 et 5.7 présentant les résultats d'implémentations matérielles du pipeline complet.

Pour résumer, la HDR-MMU est capable de capturer et de stocker le flux de pixels en cours de transfert issu du capteur. De plus, elle délivre simultanément  $P - 1$  flux de pixels précédemment mémorisés au processus de création HDR. Les signaux de lecture de BRAMs sont synchronisés avec les synchronisations verticales et horizontales du cap-



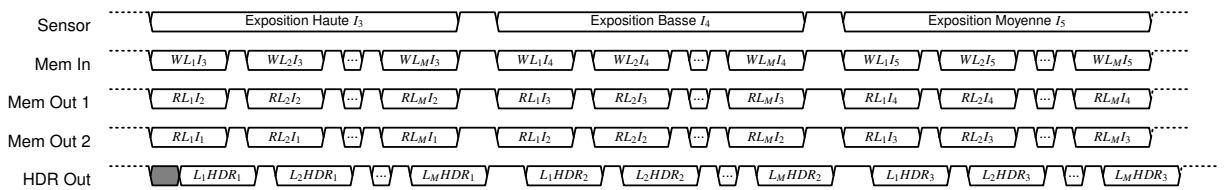
**FIGURE 4.8** – Unité de gestion de mémoire avec l'acquisition nouvelles lignes de pixels et la lecture en parallèle des lignes précédemment stockées.

teur. Avec une telle gestion de la mémoire, nous évitons d'attendre la capture des  $P$  images avant de pouvoir commencer à calculer les données HDRs. Une fois l'initialisation terminée, notre système est synchronisé avec le framerate capteur (soit 60 images par seconde), et le processus de "memory swapping" peut se dérouler tout au long des prochaines réceptions de trames vidéos. Il sera ainsi possible de produire une nouvelle image HDR à chaque nouvelle image issue du capteur.

Par ailleurs, en termes d'espace mémoire, la MMU nécessite d'enregistrer seulement  $P - 1$  trames vidéos. Le calcul d'une ligne HDR nécessite  $P$  accès mémoire (une écriture et  $P - 1$  lectures pendant l'interligne). La technique traditionnelle nécessite d'enregistrer  $P$  trames vidéos en mémoire. Il y a besoin d'au moins  $P + 1$  accès mémoire (une écriture et  $P$  opérations de lecture). Enfin, pour générer une image HDR, la MMU économise  $M$  accès mémoire en écriture (avec  $M$ , le nombre de lignes qui composent l'image).

## 4.5/ ALGORITHMIE

Un pipeline de test est réalisé afin de pouvoir quantifier qualitativement et rapidement, une série d'algorithmes dédiés HDRs. Un schéma montrant le principe de description



**FIGURE 4.9 – HDR-MMU.** Gestion de trois flux vidéos parallèles des images  $I_1$ ,  $I_2$  et  $I_3$  d'exposition basse, moyenne et haute dans le cas où  $P = 3$ . La sortie HDR est sujette à une latence algorithmique (partie grisée).

des algorithmes choisis est visible sur la Figure 4.10. L'environnement de simulation a aussi pour utilité d'aider progressivement à tester les algorithmes au cours des différentes étapes de conception. L'idée de départ est de sélectionner des algorithmes susceptibles d'être implémentables en hardware, de les retranscrire en software avec Matlab, ou bien dans un langage de plus bas niveau de type C/C++. Le but final est l'implémentation hardware, via un débogage dans l'environnement de simulation Modelsim. Chaque environnement de test permet d'avoir différentes descriptions pour un même algorithme, facilitant ainsi les comparaisons et le debug.

L'étape Matlab permet de tester un grand nombre d'algorithmes rapidement et de façon purement qualitative. Le passage en C++ a pour but de "disséquer" et d'étudier la faisabilité hardware des algorithmes sélectionnés sous Matlab. En effet, certaines description Matlab étant d'assez haut niveau, il n'est pas toujours possible de distinguer clairement au premier coup d'œil la complexité de l'algorithme en question. L'environnement C++ a en plus permis la comparaison des résultats de simulation hardware sous Modelsim avec les résultats software, dans le but d'envisager un débogage rapide et efficace si des problèmes dans le résultat visuel venaient à survenir.

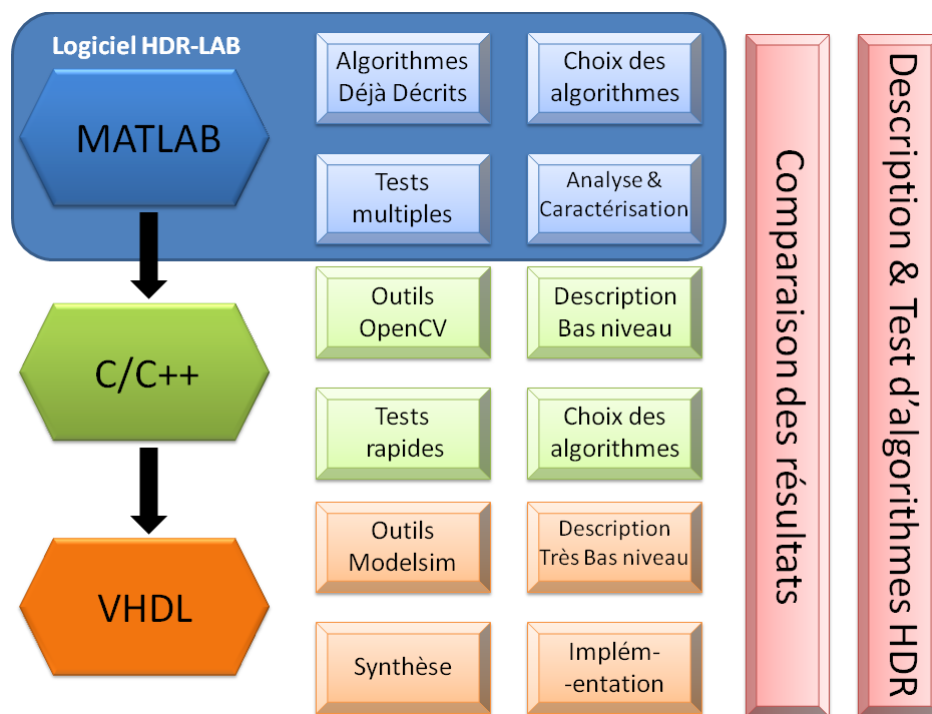
#### 4.5.1/ UN OUTIL MATLAB : *HDR-Lab*

Nous avons développé *HDR-Lab* avec l'aide d'un étudiant malaisien de l'université de Petronas. Ce logiciel dédié aux tests d'algorithmes HDRs sur Matlab, a pour but de regrouper des techniques de génération de contenu à grande dynamique, dans l'optique de pouvoir quantifier la qualité des résultats, d'effectuer des comparaisons entre algorithmes, d'analyser statistiquement les images HDRs ou d'inclure des outils dédiés HDR existants, etc.

Le programme propose de nombreuses fonctionnalités, autour de la mise en place du pipeline HDR complet, partant du choix des images jusqu'à la visualisation du résultats, en passant par l'analyse statistique, la comparaison d'images, etc. Voici une liste non-exhaustive des fonctionnalités que l'on peut y trouver :

1. Sélection d'images LDR en entrée
2. Détection ou sélection des temps d'exposition des images en entrée





**FIGURE 4.10** – Schéma global de l'organisation de test des algorithmes HDRs.

3. Sélection des paramètres pour la génération HDR ou la fusion d'exposition
4. Possibilité de pouvoir choisir parmi sept algorithmes de tone mapping et de pouvoir sélectionner leurs paramètres propres
5. Visualisation du résultat HDRi après tone mapping
6. Analyse du résultat et comparaison avec une autre image dite de "vérité terrain" (ou "gold standard"). Mise en place de métriques d'évaluation de qualité d'image (PSNR, UQI, SSIM, etc.)
7. Présentation des résultats : histogrammes, courbe de réponse, image des contributions, image des radiances en fausses couleurs.
8. Exportation et enregistrement des résultats sous Excel.

La fenêtre principale du logiciel HDR-Lab est représentée en Annexe B.1.

Le nombre d'images LDRs utilisées en entrée du pipeline HDR est compris entre 2 et 5. Nous nous limitons à 5 puisque il serait difficilement possible de réaliser un système vidéo temps réel à plus de 5 images avec la solution multiple expositions envisagée. En effet, les d'artefacts temporels qui apparaissent dans les scènes dynamiques, tels que le flicker ou les ghosts, seraient d'intensité trop importante, ce qui aurait tendance à rendre les scènes vidéos inexploitable. De plus, la majorité des systèmes de vision HDR existants n'utilisent que très rarement plus de trois images 3.2.

Toutes les comparaisons de métriques de qualité que nous ferons dans la suite du manuscrit seront réalisées avec l'étalon or, tone mappé avec l'algorithme de Drago et al. [13],

l'image HDR étant créée à partir de l'algorithme de Debevec et des 16 images LDRs du mémorial [10].

HDR-Lab est un outil qui peut être utilisé pour aider n'importe quelle application qui implémenterait le pipeline HDR complet au sein d'un système. Dans cet outil, les algorithmes les plus courants, les plus populaires, et les plus adaptés à être implémentés en hardware sont présents. Le logiciel est organisé en plusieurs phases que nous détaillerons dans les sections suivantes.

#### 4.5.1.1/ GÉNÉRATION HDR

La fonction qui est développée dans cette phase sera impliquée dans la deuxième étape du pipeline HDR. Avec l'aide des travaux de Debevec et al. [10], le script a pour but de récupérer les valeurs de luminances des scènes avec des temps d'exposition variés. Les images sont fournies par l'utilisateur par l'intermédiaire d'une interface simple. Elles doivent être parfaitement alignées les unes avec les autres, et être chacune associée à un temps d'intégration. Le script va d'abord trouver la courbe de réponse inverse  $g$  du système, pouvant être obtenue en prenant des pixels à  $x$  positions aléatoires dans les images, et en établissant les équations quadratiques (voir la Section 4.5.2.2 pour plus de détails). Une carte de radiance est ensuite construite. Nous avons rajouté la possibilité de sélectionner les paramètres relatifs à la génération HDR, tels que le nombre  $x$  d'échantillons pixeliques à utiliser pour retrouver la courbe de réponse, ou la valeur du coefficient de lissage  $\lambda$  de la courbe de réponse.

Une fois la carte de radiance enregistrée dans le logiciel, nous avons la possibilité d'effectuer le tone mapping pour visualiser le résultat.

#### 4.5.1.2/ DÉPLOIEMENT DE TMOs

Dans cette phase, sept algorithmes de tone mapping ont été inclus, dont les trois opérateurs de tone mapping les plus courants, qui sont Drago TMO [13], Duan TMO [50] et Reinhard TMO [14].

**1) TMO Drago [13] :** C'est une technique de tone mapping rapide et de haute qualité pour afficher des images à contrastes élevés sur les appareils avec une plage dynamique limitée. La fonction développée prendra en entrée le rayonnement  $E$ , retrouvé à partir de la méthode de création HDR.

**2) TMO Duan [50] :** Une nouvelle méthode de réglage d'histogramme pour afficher une image à plage dynamique élevée. Deux versions de l'algorithme sont disponibles : une version globale et une version locale. Nous n'étudierons que la version globale. Elle ne s'appuie que sur une compression logarithmique, faisant intervenir le minimum et le maximum de luminance de l'image HDR.

**3) TMO Reinhard [14] :** Cette technique s'appuie sur la pratique de la photographie. Dans notre outil, l'algorithme est décliné en trois parties correspondantes aux versions globale, globale + locale, et vidéo. La version vidéo a été réalisée grâce à une autre revue de Reinhard et al. [91]. Une technique appelée "automatic dodging and burning" est incluse dans la déclinaison global+local, mais pas pour les autres.

Une fois l'algorithme et les paramètres relatifs aux TMOs sélectionnés, la visualisation du résultat est disponible. On peut alors effectuer des tests, comparaisons et évaluation de qualité grâce à l'implantation de métriques dédiées.

#### 4.5.1.3/ MÉTRIQUES DE QUALITÉ D'IMAGE

Nous avons sélectionné six facteurs de qualité ou métriques, susceptibles de nous aider dans nos futurs choix algorithmiques. L'utilisation de ces métriques se fera dans les Sections 4.5.2 et 4.5.3. Dans tout les cas de figure, un étalon or est nécessaire afin d'effectuer les mesures.

**1) et 2) Erreur quadratique moyenne (MSE pour "Mean Squared Error") et erreur quadratique moyenne normalisée (NRMSE pour "Normalized Root Mean Square Error") :** Le MSE est couramment utilisé pour de nombreuses raisons. Il est en effet simple et facile à calculer. C'est un indicateur qui quantifie l'erreur globale en faisant intervenir la somme normalisée des erreurs individuelles mises au carré. Il permet d'estimer la qualité des images en particulier dans les diverses applications de déconvolution numérique. Il regroupe les valeurs des erreurs de prédictions en une seule mesure. MSE et NRMSE sont sensiblement équivalents.

**3) Indice de la qualité (UQI pour "Universal Quality Index") :** Wang et al. [107] proposent en 2002 une nouvelle métrique ; la mesure de qualité adaptée au système visuel humain (HVS). La fonction retourne un indice de qualité global de l'image, avec une plage de valeurs comprises dans l'intervalle  $[-1, 1]$ , faisant référence au résultat d'une comparaison entre deux images.

**4) La similarité structurelle (SSIM pour "Structural SIMilarity") :** Dans un deuxième article de Wang et al. [108], une forme plus généralisée du Uqi a été proposée. Si l'une des images comparées est considérée comme d'une qualité parfaite, alors SSIM peut calculer la qualité de l'autre image. Si les deux images sont les mêmes, alors  $SSIM = 1$ .

**5) Prédiction de différence visuelle (HDR-VDP pour "Visible Difference Metric for High Dynamic Range Images") :** L'institut Max Planck a élaboré une métrique globale dédiée aux images HDRs [109]. La métrique prédit si les différences entre deux images sont visibles à l'observateur humain ou non. Ces mesures sont utilisées pour tester :

- La visibilité de l'information, dans le but de pouvoir repérer l'information visuelle importante d'une image

- La visibilité du bruit, afin de s'assurer si nous ne percevons pas trop de distorsions dans une image. Elle a la particularité de pouvoir fonctionner à la fois avec les valeurs réelles de luminances des images HDRs, et avec les valeurs de pixels des images tone mappées. Cette métrique est aussi utilisée pour juger de la qualité d'une image après compression avec perte.

Des résultats visuels de l'application de cette métrique seront représentés sur la Figure 4.19. Nous les détaillerons lors du choix des algorithmes.

**6) Pic Signal to Noise Ratio (PSNR) :** Le PSNR est donné en unités décibels ( $dB$ ), qui mesure le rapport entre le signal de crête et la différence entre les deux images bruitée et non bruitée. Une augmentation de  $20dB$  correspond à une division par dix de l'écart quadratique entre les deux images.

$$PSNR = 10. \log \frac{255^2.M.L}{\sum_{i,j} (Xo_{ij} - Xb_{ij})^2}, \quad (4.12)$$

où  $Xo_{ij}$  et  $Xb_{ij}$  sont les valeurs de pixel à la position  $ij$  pour l'image de référence et l'image "bruitée".

La métrique PSNR est un moyen de déterminer, dans le domaine de l'image, la qualité d'une image compressée. Le niveau de distorsion est calculé à partir d'une image source. C'est une mesure largement utilisée dans la compression d'image, pour qualifier un encodeur par exemple. La SSIM, qui fait référence à la similarité structurelle des images, est censée être plus représentative de la quantification comparée au PSNR, et souvent jugée meilleure. A titre d'exemple, un spectateur trouvera plus agréable à regarder une scène dont l'objet qui attire le plus son attention est le plus net, même si le reste de l'image est bruité (PSNR faible).

La mesure de qualité d'une image est un problème assez dur à résoudre, du fait que la complexité de la perception visuelle humaine, qui est difficilement synthétisable en une seule métrique.

Nous avons choisi ces six estimateurs dans le but d'être exhaustif en termes de métrique de qualité. Chaque métrique ayant ses avantages et ses inconvénients, il est difficile de trouver une méthode qui soit objectivement la meilleure. De plus, la littérature montre qu'il est encore difficile de juger efficacement de la qualité d'une image en général, et qu'il reste des efforts à faire dans ce domaine de recherche.

#### 4.5.1.4/ OUTILS DE VISUALISATION

##### Image de contribution :

Deux utilitaires de visualisation supplémentaires ont été inclus dans le logiciel HDR-Lab. L'image de contribution est un outil qui sert à la visualisation du degré de contribution

qu'apporte les images LDRs dans la représentation finale de la carte de radiance. Un exemple de ces représentations est donné dans la Figure C.1 en annexe.

### Image fausses couleurs :

D'autre part, la carte de radiance en fausses couleurs, calculable à partir de la carte de radiance HDR, est en fait une manière de représenter la gamme de luminances par l'intermédiaire du spectre RGB entier. Cette technique permet d'associer à une plage de luminances donnée une couleur spécifique dans le but de rendre les zones correspondantes plus marquées. On peut ainsi juger plus objectivement des détails capturés dans l'image HDR qu'avec la méthode de tone mapping.

## 4.5.2/ CHOIX ALGORITHMIQUE POUR LA GÉNÉRATION HDR

### 4.5.2.1/ FORMAT DE DONNÉE

Le passage des valeurs de pixel des différentes images dans le domaine de radiance lors de la génération HDR, les calculs logarithmiques et la compression de dynamique, imposent la manipulation des nombres à virgule (ou nombres décimaux) au sein de notre architecture. À partir de là, plusieurs possibilités s'offrent à nous. Nous devons prendre en compte le degré de précision nécessaire à nos calculs, connaître approximativement la plage des valeurs que nous utiliserons, etc. Le format à virgule fixe et le format flottant seront envisagés. Voici une description de ces formats, sur lesquels nous nous sommes appuyés pour justifier le choix qui a été fait pour notre système.

#### 1. Le format flottant IEEE754

La dynamique étendue et la précision de l'arithmétique flottante sont parfois requises dans certaines applications, telles que dans les communications sans fil, le domaine du traitement du signal, de l'audio, le contrôle qualité, l'instrumentation, les applications de traitement d'images, etc.. Le standard IEEE754 à simple précision est la représentation d'un nombre flottant stocké dans un mot de 32 bits, composé d'1 bit de signe (s), 8 bits pour l'exposant et 23 bits de mantisse (Voir Tableau 4.1).

s	exposant	mantisse
0	0111 1110	111 1111 1111 1111 1111 1111

**TABLE 4.1** – Représentation du nombre flottant 0.99999994 en format IEEE754 simple précision

Étant donné que les nombres à virgule flottante sont répartis sur un espace infini de réels, et que l'on cherche à les mapper sur un nombre fini de bits (32 bits pour une simple précision IEEE754), certaines erreurs peuvent apparaître et être tolérées. En flottant, il y a trois sources principales d'erreurs : l'arrondi, l'underflow et le débordement. L'underflow intervient lorsque l'opération génère un nombre proche de zéro qui est trop petit pour être représenté avec la précision sélectionnée. L'over-

flow provient d'un nombre trop grand, et il en résulte un nombre représentant l'infini correctement signé.

Une étude [110] fait l'état de l'art des méthodes qui implémentent des calculs flottants sur FPGA. Les travaux mettent en avant des descriptions d'opérateurs et des optimisations hardware relatives aux contraintes et aux besoins des FPGAs. Nous citerons, à titre indicatif, quelques exemples :

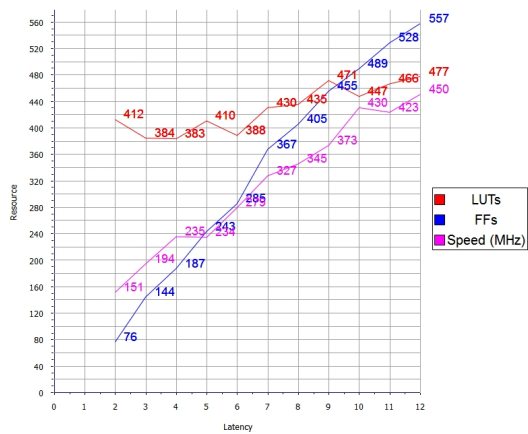
- La corrections d'arrondis dans les multiplications [111]
- L'optimisations de la précision
- La parallélisation des opérations [112]
- Les designs de fonctions trigonométriques [113]
- L'implémentation paramétrable de l'exponentielle et du logarithme [114]
- Les multiplications de matrices [115]
- Le développement d'opérateurs paramétrables [116]
- La multiplication par une constante [117]
- La division itérative [118], la réciproque de la racine carré [119], etc.

En exploitant au mieux les capacités du FPGA, il est dit que les performances peuvent même dépasser celles des processeurs en termes de rapidité d'exécution et de précision de calcul.

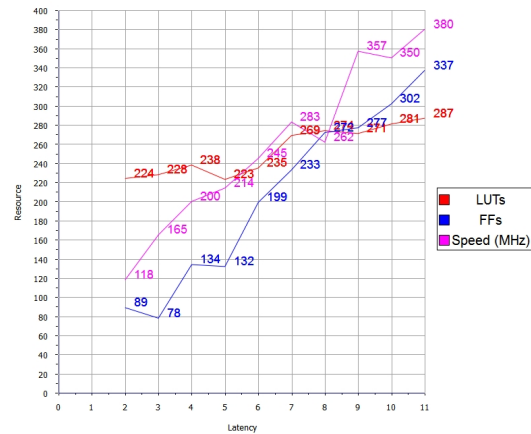
Grâce à la particularité des FPGAs d'être massivement parallélisables, et à l'augmentation constante de leur capacité, il devient possible d'implémenter des opérateurs flottants avec plus de puissance de calcul qu'un processeur à algorithmie simple ou double précision. Il est aussi possible de paramétrer la latence des opérateurs, ce qui permet de pouvoir trouver un bon compromis entre délai d'exécution (latence faible), ressources utilisées (nombre de slices, Flip-Flops, DSPs, etc.) et vitesse d'exécution (latence forte). Une vue du tableau de bord et des différents paramètres d'un opérateur flottant de chez Xilinx est présent sur la Figure 4.13. A titre informatif, une synthèse de l'utilisation de ressources sur Virtex 6 pour les opérateurs additionneur/multiplicateur est donnée dans la Figure 4.11. On y voit les courbes correspondant aux opérateurs implémentés de différentes manières ("high speed", pas d'utilisation de DSPs, "low latency"). On remarque par ailleurs que plus la latence augmente, plus la fréquence maximale de calcul peut augmenter.

## 2. Le format à virgule fixe

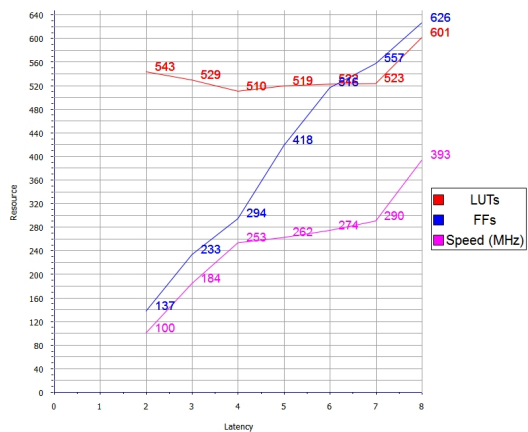
Le codage en virgule fixe demande de respecter certaines règles. Il faut définir pour chaque donnée, la position de la virgule, afin de séparer la partie fractionnaire de la partie fixe. Il est nécessaire de faire attention à ce que les algorithmes ne provoquent pas de débordements, contrairement au codage flottant. Ce codage est donc utile lorsque l'on travaille sur des nombres s'étalant sur une dynamique assez restreinte, et pour lesquels on peut définir une précision fixe de part et d'autre des opérateurs. Il est alors nécessaire de regrouper les données de façon à disposer et à harmoniser les données par blocs. Ces blocs auront, soit toutes les données avec



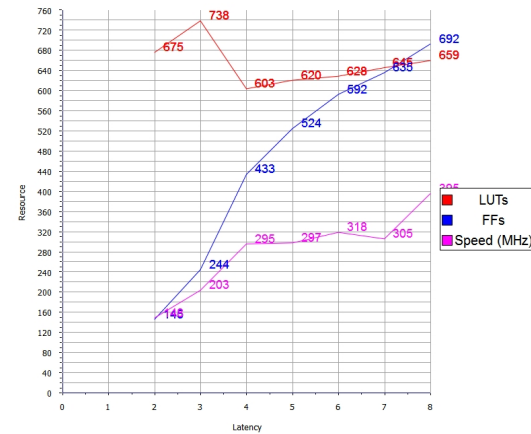
(a) Add high speed, no DSPs usage.



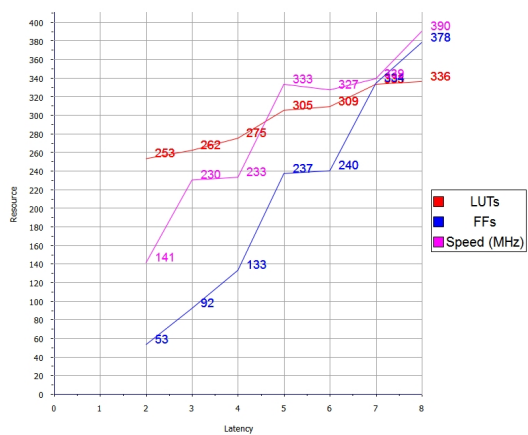
(b) Add high speed, full DSPs usage.



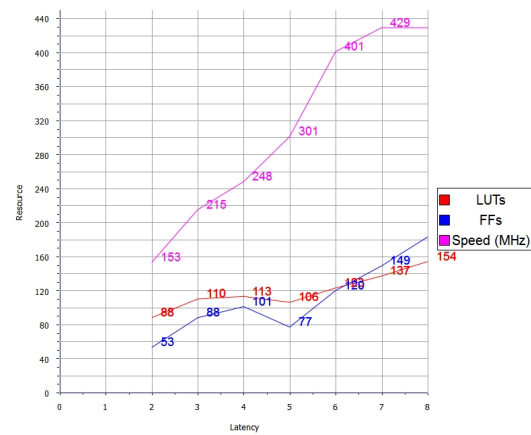
(c) Add low latency, no DSPs usage.



(d) Mult. high speed, no DSPs usage.



(e) Mult. high speed, med DSPs usage.



(f) Mult. high speed, full DSPs usage.

**FIGURE 4.11** – Utilisation des ressources pour les opérateurs d'addition et de multiplication sur Virtex 6.

la même largeur, soit des données regroupées en sous-blocs de même largeur, soit toutes les données ayant leur propre largeur. La difficulté réside dans le fait de prévoir la dynamique de calcul, et d'interconnecter toutes les données entre elles.

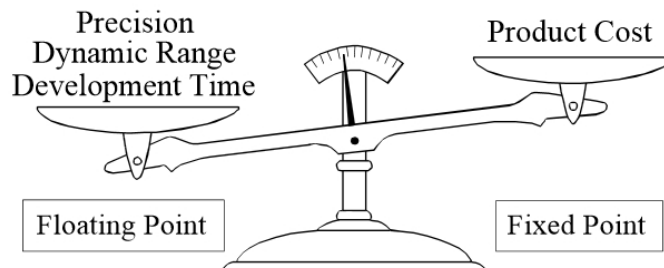


FIGURE 4.12 – Illustration tirée du livre de Steven et al. [120].

### 3. Comparaison

Une implémentation issue de l'état de l'art [121], montre des résultats d'estimation de surface et de délai pour des opérateurs fixes et flottants. Ils donnent par ailleurs des données chiffrées pour les fonctions  $\sin \frac{\pi}{4}x$  et  $\log_2(1+x)$  sur un intervalle  $x$  défini sur  $[0, 1[$ , et ce pour plusieurs ordres d'approximation. Ils remarquent que la surface augmente exponentiellement avec la précision (nombre de bits de représentation). De plus, les délais de sortie augmentent linéairement avec la précision. Par exemple, une précision de 32 bits sur l'opérateur en virgule fixe de type sinus requiert environ 1827 slices pour un délai de 39ns. Pour comparaison avec les algorithmes présents dans le LogiCORE de Xilinx, on peut considérer l'IP Cordic de Xilinx implémentant également le sinus en virgule fixe. Une option dite "fully parallel" permet une sortie en l'espace d'un cycle d'horloge seulement (5ns pour une horloge à 200MHz), mais consommant des ressources de l'ordre de 1983 slices sur Spartan-3A. L'autre option dite "word serial" permet d'avoir une sortie avec latence sur une surface très réduite de silicium. On pourra, par exemple, implémenter la fonction sinus 32 bits avec 653 slices avec une fréquence maximum de 86MHz. Au vue de ces résultats, il a été intéressant de considérer plus particulièrement les implémentations fournies par Xilinx, notamment en se penchant sur la déclinaison en virgule flottante des opérateurs. Le choix d'utiliser les algorithmes flottants s'est fait spontanément, au vu de la dynamique énorme résultant des calculs de radiance, et en considérant que l'architecture en flottants ne consomme pas énormément plus de ressources que l'architecture en virgule fixe. De plus, les contraintes de début de thèse n'étaient pas de réaliser un système qui soit le plus minimaliste possible, mais qui soit plutôt axé temps-réel. Nous verrons dans le Chapitre 5, que ce choix fût judicieux, de part la qualité des images produites à faible bruit de quantification, et aux faibles ressources surfaciques finalement empruntées.

À titre comparatif, sur Virtex 6, un multiplieur en virgule fixe  $24 \times 24$  en utilisant les LUTs comme ressources, demande 624 LUT6s. Pour le flottant, un multiplieur



simple précision demande 659 LUTs. On peut voir que l'écart est très faible pour ce type d'opérateur. Le tableau sur la Figure 4.2 présente quelques comparaisons d'utilisation des ressources pour les deux systèmes de représentation des nombres sur Virtex 6. La dynamique des opérateurs fixes est de 24 bits, et la représentation des nombres en flottants est la simple précision (32 bits). La version 5.0 de l'IP de Xilinx pour les nombre flottants a été testée. Pour les opérateurs d'addition/-soustraction et de multiplication, nous définissons deux types d'architectures, une uniquement basée sur la logique, et une autre utilisant des blocs DSPs. On peut remarquer que la représentation en virgule fixe est moins gourmande en ressources uniquement pour des opérations simples de types addition/soustraction. De par ce tableau, on peut nettement mieux envisager d'utiliser l'arithmétique flottante, utilisant finalement assez peu de ressources. Depuis peu, l'utilisation de DSPs est possible sur FPGA dans le cas d'implantation de certains opérateurs flottants, ceci diminuant de façon drastique la quantité de ressources logiques utilisées.

Opérateur	Virgule Fixe		Flottant	
	LUTs	DSPs	LUTs	DSPs
Additionneur/soustracteur	75	0	477	0
	0	1	287	2
Multiplicateur	696	0	659	0
	132	1	107	3
Diviseur 1 Cycle	1377 (Radix-2)	0	780	0
Diviseur 25 Cycles	-	-	187	0
Racine carrée 1 Cycle	1550	0	533	1
Racine carrée 25 Cycles	-	-	170	1

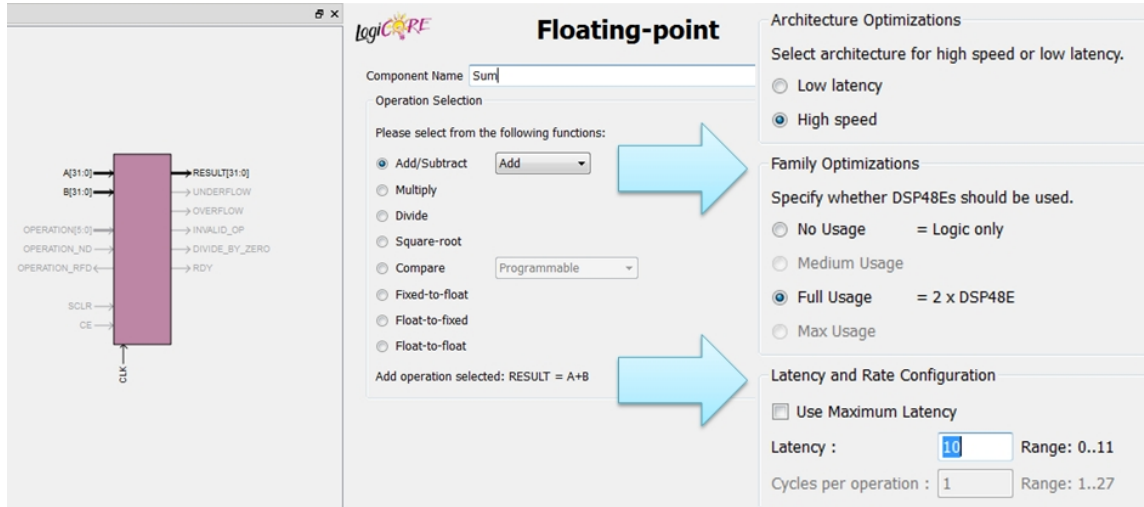
**TABLE 4.2** – Comparaison de l'estimation des ressources des opérateurs arithmétiques sur Virtex 6, pour une représentation des nombres en virgule fixe et flottant.

Dans notre cas, il est assez difficile d'évaluer à première vue la dynamique de la radiance calculée directement. En effet, celle-ci dépend de la valeur des temps d'exposition des pixels à fusionner.

Au vu de la gamme dynamique qui sera utilisée, nous utilisons les opérateurs flottants inclus dans les logiciels Xilinx, nommés "Floating-Point Operator", dont l'implémentation est facilitée et adaptée aux FPGAs. Ces IPs sont configurables et peuvent être ciblées sur un nombre important de plates-formes.

#### 4.5.2.2/ DEBEVEC ET AL.

Les appareils photographiques numériques ont une gamme dynamique limitée puisqu'ils ne peuvent capturer des images que sur une gamme de 8 à 14-bit. La gamme dynamique des scènes réelles n'est donc pas totalement récupérée. Si l'appareil photo a une



**FIGURE 4.13** – Tableau de bord du LogiCORE de chez Xilinx. Implantation d'un composant arithmétique en flottant IEEE754. Sélection de l'opérateur et optimisations.

réponse linéaire, on peut facilement récupérer la luminance HDR  $E_{ij}$  à partir de chaque luminance  $Z_{ij}^p$  et des valeurs des durées d'exposition  $\Delta t_p$  pour chaque trame  $p$ . Malheureusement, les caméras n'ont pas une réponse linéaire et nous devons estimer la fonction de transfert non linéaire de la caméra appelée  $f$ , afin de combiner correctement les différentes expositions.

Quatre algorithmes populaires pour la récupération de cette fonction de transfert sont extraits de la littérature : Debevec et al. [10], Mitsunaga et al. [12], Mann et al. [23] et Robertson et al. [11], et ont été décrits dans la Section 2.3.1. Selon la description détaillée de ces méthodologies [122], et la comparaison de leurs implémentations logicielles en temps réel [42], nous avons décidé d'utiliser la méthode de Debevec :

$$Z_{ij}^p = f(E_{ij}\Delta t_p). \quad (4.13)$$

La fonction  $g$  est définie comme l'inverse de la courbe de réponse du système  $g = \ln f^{-1}$  et peut être obtenue par minimisation de la fonction suivante :

$$O = \sum_{i=1}^M \sum_{j=1}^N \sum_{p=1}^P [g(Z_{ij}^p) - \ln E_{ij} - \ln \Delta t_p]^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} g''(z)^2, \quad (4.14)$$

où  $\lambda$  est un scalaire de pondération en fonction de la quantité de bruit attendu sur  $g$ ,  $Z_{min}$  et  $Z_{max}$  sont respectivement la plus basse et la plus grande des valeurs de pixels. L'évaluation de  $g$  requiert une évaluation d'un nombre fini de valeurs comprises entre  $Z_{min}$  et  $Z_{max}$  (1024 valeurs par exemple pour un capteur de précision 10-bit). Ces valeurs peuvent être évaluées de façon préliminaire à partir d'une séquence de plusieurs images, puis stockées dans l'appareil, et réutilisées pour convertir les valeurs de pixels en radiance. Pour récupérer la valeur de radiance HDR d'un pixel particulier, toutes les expositions

disponibles de ce pixel sont combinées en utilisant l'équation suivante :

$$\ln E_{ij} = \frac{\sum_{p=1}^P \omega(Z_{ij}^p) [g(Z_{ij}^p) - \ln \Delta t_p]}{\sum_{p=1}^P \omega(Z_{ij}^p)}, \quad (4.15)$$

où  $\omega(z)$  est la fonction de pondération de Debevec qui donne plus de poids aux valeurs de pixels les plus proches du milieu de la gamme dynamique du capteur, comme décrits dans la Section 4.5.2.2 :

$$\omega(z) = \begin{cases} z - Z_{min} & \text{for } z \leq \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - z & \text{for } z > \frac{1}{2}(Z_{min} + Z_{max}) \end{cases} \quad (4.16)$$

Comme le disent Tocci et al. [82], une telle méthode est largement utilisée pour produire du contenu vidéo HDR, en capturant des images avec une alternance d'expositions lumineuses et sombres. Pour notre part, nous décidons d'utiliser cette méthode, car elle s'est avérée être très robuste et ne nécessitant que peu de ressources.

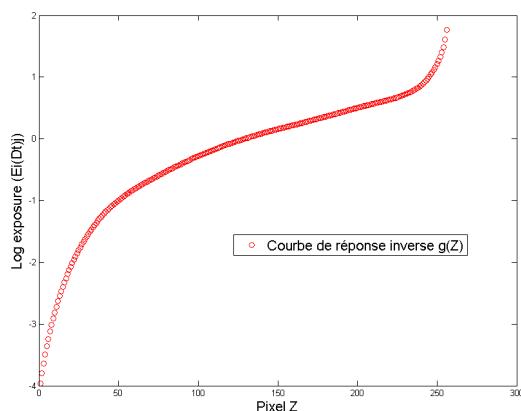
Afin de satisfaire à la fois les contraintes temporelles et les exigences matérielles, certaines simplifications d'algorithmes décrites dans les Sections 4.5.2.3 et 4.5.2.5 sont proposées.

#### 4.5.2.3/ DÉTERMINATION DE LA COURBE DE RÉPONSE

La quantité numérique d'un pixel est en relation directe avec la quantité de lumière frappant l'élément du capteur pendant l'exposition. Seulement la sortie numérique varie de façon non linéaire avec la quantité de lumière. Il est alors nécessaire de déterminer la courbe de réponse de notre système ; c'est à dire la relation liant la valeur de radiance réelle et la valeur de pixel en fonction de l'exposition. Comme inscrit dans la revue de Debevec et al. [10], un minimum de deux expositions est suffisant pour retrouver la courbe de réponse, si l'information présente sur les deux photographies permet de relier les deux expositions. Néanmoins, un nombre plus élevé permet d'augmenter le rapport signal sur bruit pendant la récupération de la courbe. Pour notre part, nous utilisons cinq images pour retrouver la courbe de réponse d'une manière totalement orientée software (voir Figure 4.14). Les images sont récupérées depuis notre caméra via RS232 ou Ethernet. Il faut veiller à ce que les échantillons soient alignés, sinon l'estimation de la courbe et les correspondances faites entre les positions de pixels sont faussées. Puisque la génération de la courbe nécessite la connaissance précise des temps d'exposition des images LDRs capturées, nous avons implémenté la possibilité de configurer les temps d'exposition de notre capteur. Cette courbe servira pour les futures générations d'images HDRs faites avec notre caméra.

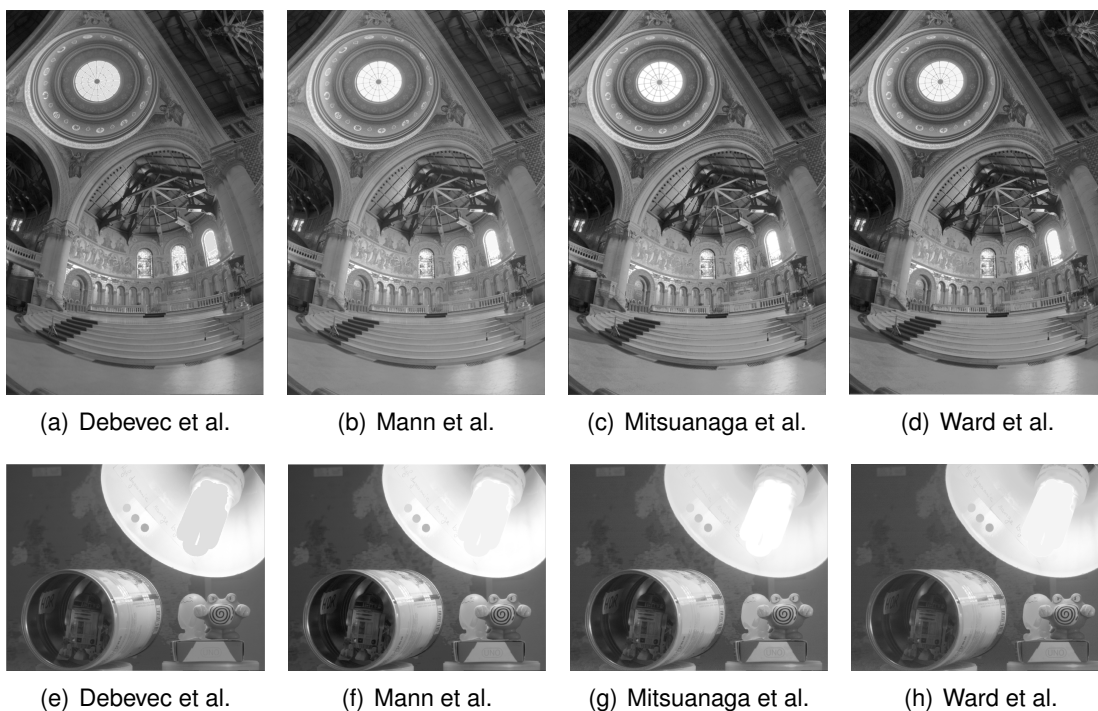
Les images utilisées pour la récupération de la courbe  $g$  sont illustrées dans la Figure en

## Annexe C.1.



**FIGURE 4.14** – Courbe de réponse inverse  $g(Z_{ij}) = \ln(E_i \cdot \Delta t_j)$  de notre système HDR-ARtiSt, créée à partir de 5 images LDRs et de l'algorithme de Debevec. Voir l'Annexe C pour plus de détails.

## 4.5.2.4/ CHOIX DE LA FONCTION DE PONDÉRATION



**FIGURE 4.15** – Les quatre fonctions de pondération en test. l'algorithme de visualisation tone mapping est celui de Duan et al. [50].

La Figure 4.15 met en évidence les quatre fonctions de pondération décrites dans le Chapitre 2 (voir notamment le Tableau 2.5). On peut remarquer qu'il n'y a pas de différences flagrantes lorsque l'on utilise indépendamment ces fonctions. Les tests en simulation software sont effectués sur l'image de Debevec et sur les images issues de la caméra

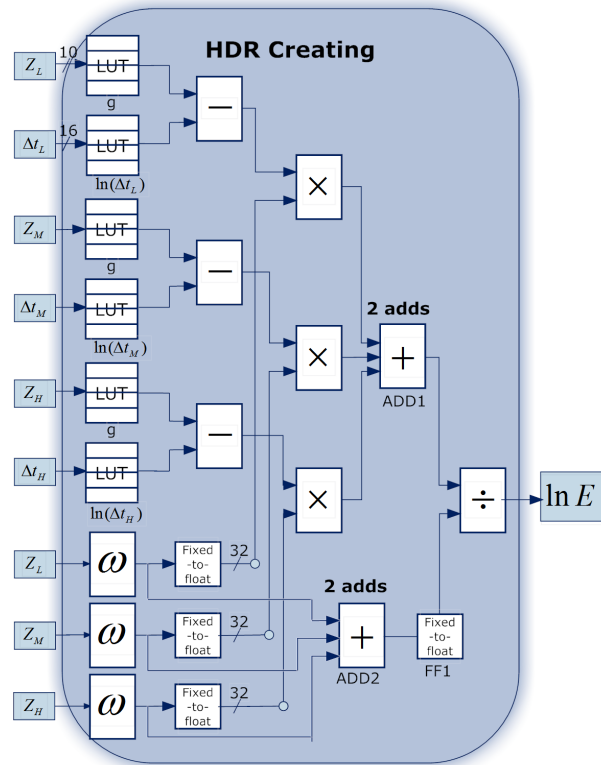
HDR-ARtiSt. Nous avons choisis les algorithmes de Debevec et Mitsunaga à implémenter sur la carte, chacun ayant ses propres avantages et inconvénients. Le problème de Debevec est que la fonction de pondération rejette totalement les pixels aux extrêmes de la dynamique de sortie du capteur, ce qui tend à rendre l'image incohérente à certains endroits, notamment pour les hautes luminosités. Néanmoins, nous avons pu supprimer cet effet à l'implémentation, en détectant lorsque tous les pixels de toutes les images à une position  $ij$  sont à 1023 ou à 0. Ce problème est totalement effacé en software avec la pondération de Mitsunaga, au détriment d'un bruit plus présent dans l'image résultante.

#### 4.5.2.5/ SIMPLIFICATIONS

Le calcul de la fonction  $g$  n'a pas été implémenté sur la plate-forme, car il n'est nécessaire de l'estimer qu'une seule fois. Ainsi, les paramètres de la courbe sont évalués de façon préliminaire sur PC, à partir d'une séquence d'images. Une fois que toutes les valeurs sont calculées, tel qu'on le ferait avec une courbe de correction gamma, la courbe de réponse est stockée dans une "lookup table". La LUT est construite avec l'utilitaire "Block Memory Generator" fourni par Xilinx. On crée un fichier .COE d'initialisation du contenu de la ROM générée, regroupant les valeurs hexadécimale de la courbe. De plus, afin de réduire la complexité de calcul et d'optimiser les ressources matérielles, d'autres opérations mathématiques, telles que les logarithmes népériens, sont également pré-calculées et enregistrées dans des LUTs.

Finalement, le coût matériel de l'application de création HDR ne dépend que du nombre de trames utilisées pour la reconstitution d'une image vidéo HDR (voir Figure 4.16 pour la configuration HDR-3). Chaque trame nécessite deux opérateurs arithmétiques de 32-bit (1 soustracteur et 1 multiplicateur) et une transition à partir du format 10-bit en *IEEE754* 32-bits de largeur (*Fixed-to-Float*). Une structure en pipeline est utilisée pour l'accélération du traitement.

Pour une architecture à 3 images, il faut comme autres ressources matérielles, deux additionneurs 10-bit, deux additionneur 32-bit, un diviseur de 32-bit et un opérateur Float-to-Fixed. Le schéma est simplifié au niveau des additionneurs, car les additionneurs à trois entrées sont en fait composés chacun de deux additionneurs à deux entrées. Nous aurions pu effectuer les additions du bloc ADD2 en flottant directement après les blocs Fixed-to-Float pour éviter d'utiliser un autre bloc de passage Fixed-to-Float (FF1) après ADD2, mais nous nous sommes rendu compte que le fait de faire 2 additions d'entiers et un passage FF1 était moins coûteux en ressources que le fait de faire juste deux additions en flottants.



**FIGURE 4.16** – Création du pipeline HDR pour la vidéo HDR-3, en utilisant une architecture en cascade. Des LUTs permettent de stocker la courbe de réponse ainsi que des valeurs pré-calculées de gamme de logarithmes.

#### 4.5.3/ CHOIX ALGORITHMIQUE POUR LE TONE MAPPING

Pour notre application, le choix d'un opérateur de tone mapping a été fait après comparaisons et tests de plusieurs algorithmes en Matlab (HDR-Lab) et en C++. Les tests ont toutefois été axés vers du tone mapping d'image HDR construite à partir de 2 ou 3 images, configuration la plus envisagée à être implémentée sur une caméra intelligente.

Nous avons utilisé les trois images (a), (b) et (c) présentes dans la Figure 4.17, afin d'essayer de déterminer quel algorithme choisir parmi ceux que nous avons implanté en Matlab. L'image de référence est le mémorial de Debevec. Les temps d'exposition pour les trois images sont respectivement 32s, 1s et 31ms.

Afin de tester les algorithmes de tone mapping sous HDR-Lab, nous générons tout d'abord une seule image HDR qui sera utilisée en entrée des TMOs. Nous avons implanté une technique de visualisation des images HDR, appelée "false color" (voir Figure 4.17 (e)), qui attribue une couleur du spectre visible pour chaque valeur de radiance. On peut ainsi visualiser la répartition de toute la gamme dynamique de l'image HDR générée.

De plus, une donnée indicative est donnée par l'intermédiaire de l'image dite de contribution. Au vue de la fonction de pondération utilisée lors de la génération HDR, chaque pixel de chaque image LDR à une position donnée, ne contribue pas de la même manière

TMO	PSNR	UQI	SSIM	MSE	NRMSE	Temps(s)
<b>Drago et al.</b>	34.93	0.54	0.58	20.91	0.1	5.43
<b>Duan et al.</b>	42.34	0.97	0.91	3.79	0.01	5.71
<b>Reinhard et al. (local)</b>	25.72	0.93	0.81	174.13	0.68	10.45
<b>Reinhard et al. (global)</b>	29.11	0.89	0.8	79.86	0.32	5.52
<b>Schlick et al.</b>	40.71	0.43	0.56	5.53	0.03	5.50
<b>Tumblin et al.</b>	42.01	0.14	0.29	4.09	0.02	5.59

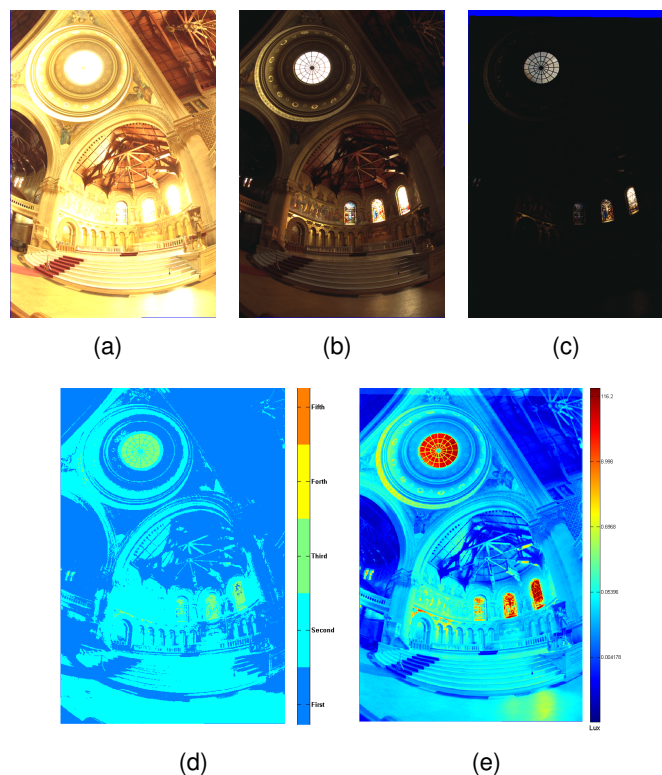
**TABLE 4.3** – Comparaisons des métriques issues du logiciel HDR-Lab, pour les algorithmes TMOs appliqués à une image HDR commune construite à partir de 3 images. Les temps d'exposition utilisés sont 32s, 1s et 31,25ms.

aux valeurs finales de luminances de l'image HDR. Par exemple, un pixel saturé blanc ou noir, ne contribuera pas autant, voire ne contribuera pas du tout à la définition de l'image HDR, et on préférera un pixel de valeur non extrême, proche du milieu de gamme (pixel proche de 128). L'image de contribution est codée sur cinq couleurs maximum, représentant les cinq expositions des cinq images LDRs en entrée. L'image de la Figure 4.17 (d) a été créée à partir de trois images, et est donc codée avec trois couleurs : bleu, bleu ciel et vert pour les images (a), (b) et (c). On remarque par exemple, que les pixels de valeurs de luminance les plus forts à l'intérieur de la coupole, sont issus principalement de la contribution de l'image (c), de valeur d'exposition la plus faible. De la même manière, les luminances reçues du transept sont définies principalement par les pixels des images d'exposition fortes.

La Figure 4.18 représente un résumé de tests de métrique issu de notre logiciel HDR-Lab. On peut voir que les algorithmes de Reinhard et al. globaux et locaux sont en général équivalents. D'autre part, on remarque que pour les métriques UQI et SSIM, l'algorithme de Duan et al. est largement au dessus. Pour les besoins de calcul, on remarque bien le surplus de temps nécessaire à la partie locale de l'algorithme de Reinhard, alors que les algorithmes globaux s'exécutent dans un intervalle de temps semblable. À première vue, nous pouvons sélectionner l'algorithme de Duan pour ses bons résultats pour UQI et SSIM, métriques relativement les plus appropriées à l'évaluation de la qualité des TMOs. De plus, le tableau 4.3 appuie encore plus ces résultats en utilisant une image HDR utilisant des images très espacées en termes d'exposition, permettant d'avoir presque toute la dynamique de l'église en trois clichés, contrairement aux résultats présents dans la Figure 4.18.

La Figure 4.19 montre les résultats de l'algorithme HDR-VDP 2 [109]. L'image de référence est l'étalon or de Drago et al. (voir Annexe E.1). L'image test est l'image HDR tone mappée avec les algorithmes cités. Cette image a été construite à partir de 3 images LDRs. La couleur rouge signifie la grande probabilité qu'aura le spectateur à distinguer une différence entre l'image de référence et l'image de test. À contrario, la couleur bleu montre une faible probabilité à percevoir une différence visuelle. Les images sont clas-





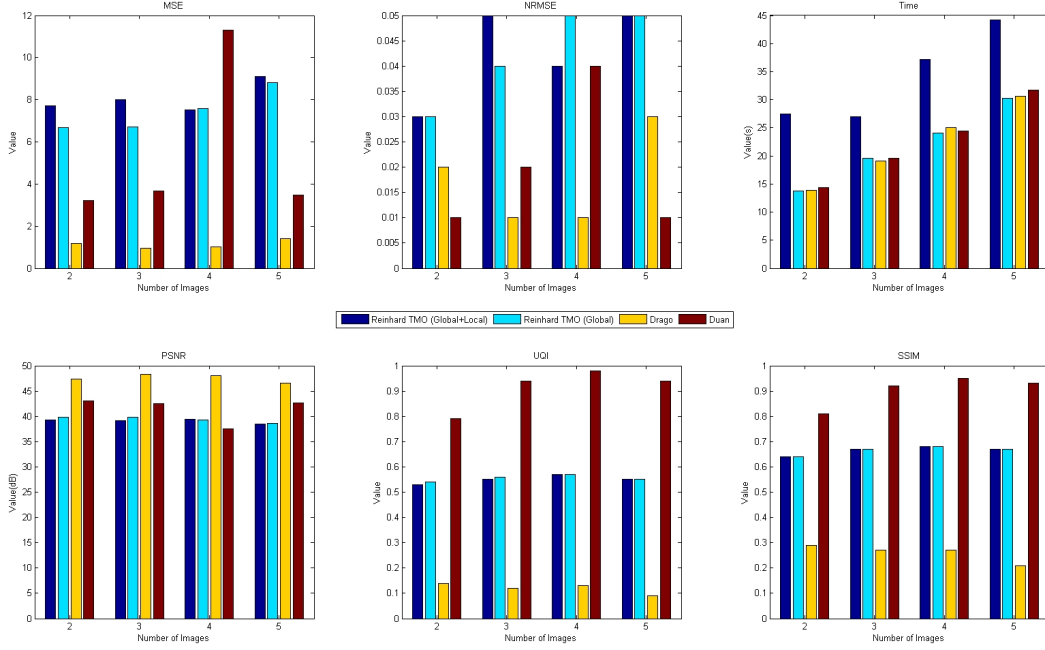
**FIGURE 4.17** – Les trois images LDRs (a), (b) et (c) qui nous ont servi à effectuer des comparaisons entre les TMOs. (d) et (e) sont respectivement l'image de contribution maximale et l'image fausse couleur des luminances de l'image HDR.

sées de l'image la moins bien notée à la mieux notée. On remarque que l'algorithme de Duan donne de bons résultats.

Les pixels HDRs sont représentés par une grande profondeur de bits, ce qui les met en conflit avec les dispositifs d'affichage standards. En ce qui concerne l'opération finale de tone mapping, celle-ci permet d'afficher sur un écran standard (avec une dynamique de 8 bits par composante RVB) des images HDRs dont la dynamique atteint plusieurs dizaines de bits. La difficulté du tone mapping est donc d'obtenir une image affichable sur un écran tout en essayant de préserver l'intégralité de l'information qui est contenue dans la carte de radiance de l'image HDR. Cadik et al. [58] montrent que la partie globale d'un opérateur de tone mapping est essentiel pour obtenir de bons résultats pour ce qui est de la restitution des contrastes et de la qualité d'image. Une expérience psychophysique par Yoshida et al. [63], basée sur une comparaison directe parmi les apparences des images HDR du monde réel montre que les méthodes globales comme Drago et al [13] ou Reinhard et al [14] sont considérées comme celles qui produisent les images les plus naturelles. En outre, une reproduction globale des tonalités est la meilleure façon d'atteindre les contraintes de temps réel, car les opérateurs locaux exigent des calculs plus complexes.

Selon nos contraintes matérielles, le meilleur compromis est le tone mappeur global de





**FIGURE 4.18** – Résultat de comparaison des algorithmes de tone mapping [13, 14, 50] sous HDR-Lab.

Duan et al. [50]. Nous avons donc choisi l’algorithme de Duan et al. [50] pour l’implémentation temps réel sur FPGA. D’après les mesures de qualité et les résultats trouvés dans les sections précédentes, il nous paraît judicieux d’utiliser cet algorithme. À titre d’illustration, l’image HDR représentée sur la Fig. 4.20 a été obtenue avec cet algorithme.

#### 4.5.3.1/ DUAN ET AL.

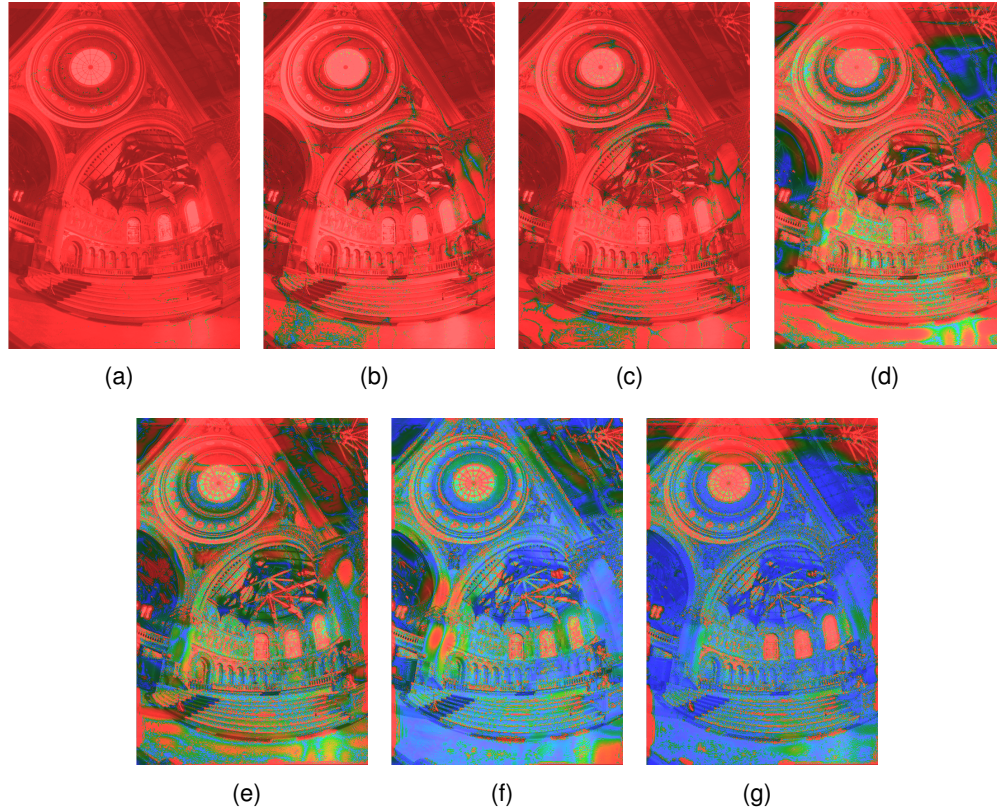
Nous verrons ici l’équation de tone mapping de l’algorithme global logarithmique de Duan et al. [50]. Le tone mappeur de Duan et al. comprime la luminance d’un pixel HDR  $E_{ij}$ , issue de la création HDR précédente, en une luminance affichable  $D_{ij}$  avec l’équation suivante :

$$D_{ij} = C * (D_{max} - D_{min}) + D_{min}$$

(4.17)

$$\text{with } C = \frac{\log(E_{ij} + \tau) - \log(E_{ij(min)} + \tau)}{\log(E_{ij(max)} + \tau) - \log(E_{ij(min)} + \tau)},$$

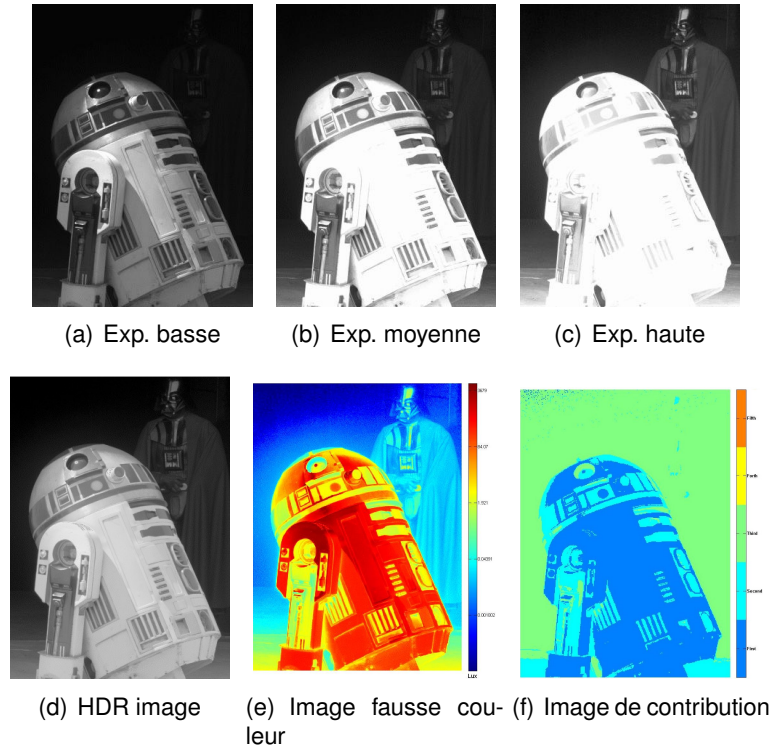
où  $E_{ij(min)}$  et  $E_{ij(max)}$  sont les luminances minimum et maximum de la scène et  $\tau$  est un paramètre de réglage de luminosité globale.



**FIGURE 4.19** – Résultats de HDR-VDP 2 [109] ("Visible Difference Metric for High Dynamic Range Images") sur les algorithmes sélectionnés dans HDR-Lab. (a) : Tumblin [45], (b) : Drago [13], (c) : Schlick [123], (d) : Reinhard local [14], (e) : Reinhard global [14], (f) "Exposure Fusion" [64], (g) : Duan [50].

#### 4.5.3.2/ SIMPLIFICATIONS

La mise en œuvre matérielle de l'équation 4.17 nécessite l'estimation préliminaire des termes  $E_{ij(min)}$  et  $E_{ij(max)}$  par rapport à l'image HDR avant toute opération de tone mapping. Cependant, comme notre pipeline matériel calcule l'image HDR et le tone mapping sur le flux entrant de pixels, une telle approche est inconcevable. Nous faisons l'hypothèse que les conditions lumineuses ne varient pas de façon significative entre les deux trames consécutives capturées à 60 fps. Donc, ces deux termes sont calculés à partir de la trame courante HDR (opérateurs  $CMPMin$  et  $CMPMax$ ), stockés dans des registres ( $Reg$ ), puis utilisés comme  $Min/Max$  pour la transformation de tone mapping dans l'image HDR suivante, comme on le voit sur la Figure 4.21.  $\tau$  contrôle la luminosité globale de l'image mappée. Selon Duan et al. [50],  $\tau$  rend l'image plus sombre si on l'augmente et l'éclaircit si on le diminue. Plusieurs valeurs ont été testées en simulation et nous avons décidé de fixer  $\tau = 0$ , car cela simplifie la mise en œuvre matérielle et donne des résultats satisfaisants. Nous avons également fixé  $D_{max} = 255$  et  $D_{min} = 0$  correspondant aux gammes dynamiques des moniteurs standards (format 8-bits par canal). Ces aspects pourront changer dans l'avenir en raison des efforts actuels dans le domaine de la



**FIGURE 4.20** – Image HDR après tone mapping de Duan et al. [50] à partir de trois images LDRs. La carte de radiance a été construite avec l’algorithme de Debevec et al. [10]. Les trois images LDRs sont tirées d’Internet.

recherche au niveau de la construction d’écrans à haute dynamique. Notre design sera en mesure de suivre cette évolution. Les images résultantes du tone mapping, avec des gammes plus importantes, pourront facilement être produites en augmentant  $D_{max}$  et en changeant la dynamique de l’opérateur Float-to-Fixe.

#### 4.5.3.3/ REINHARD ET AL.

Nous avons aussi adapté un algorithme de tone mapping global proposé par Reinhard et al. [14], choisi pour sa simplicité et sa restitution naturelle des contrastes d’images HDRs (voir l’étude de Akyüs et al. [56]). Une scène prise par bracketing doit être tone mappée en respectant les conditions lumineuses dans laquelle elle a été prise (lumière du jour ou nuit /obscurité). Reinhard et al., dans leur livre [23], ont démontré qu’il était nécessaire d’effectuer un calibrage de l’image par l’intermédiaire d’un facteur d’étalonnage noté  $a$ , dans le but d’avoir une uniformisation des valeurs relatives à une scène. On calcule d’abord la radiance moyenne de l’image HDR comme ceci :

$$\bar{E}_{ij} = \exp \left( \frac{1}{N} \sum_{i,j} \ln(E_{ij}) \right), \quad (4.18)$$

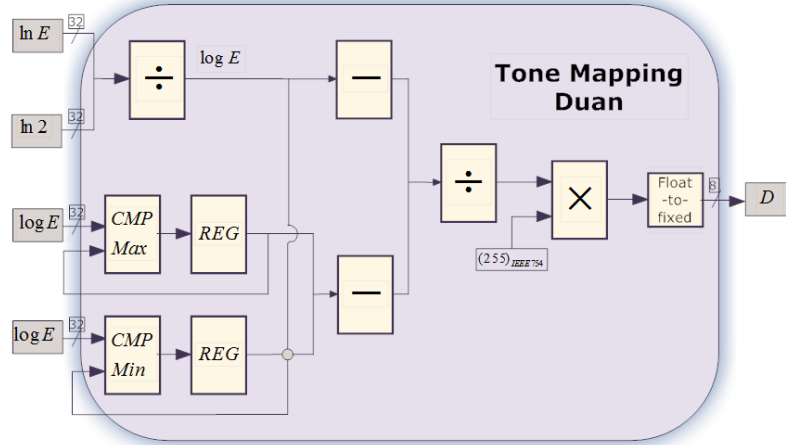


FIGURE 4.21 – Calcul de tone mapping par le méthode de Duan et al. [50].

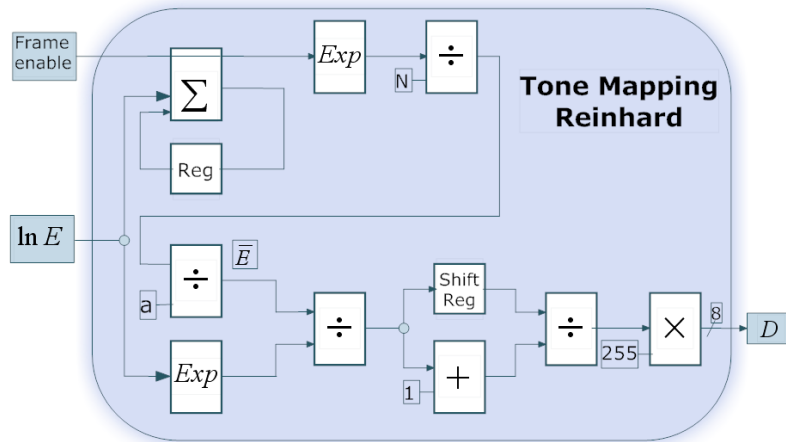


FIGURE 4.22 – Calcul de tone mapping par le méthode de Reinhard et al. [14].

où  $E(i, j)$  est la luminance du pixel à la position  $(i, j)$ ,  $N$  est le nombre total de pixels dans l'image. Ensuite, nous faisons correspondre la radiance moyenne de la scène avec la valeur grise de l'image affichable :

$$\begin{aligned}
 D_{ij} &= 255 \cdot \frac{a \frac{E_{ij}}{\bar{E}_{ij}}}{1 + a \frac{E_{ij}}{\bar{E}_{ij}}} \\
 &= 255 \cdot \frac{1}{1 + \frac{E_{ij}}{a \bar{E}_{ij}}},
 \end{aligned}
 \tag{4.19}$$

où  $a$  est la constante de calibrage estimée suivant l'illumination de la scène.

## 4.6/ CONCLUSION

Dans ce chapitre, nous avons défini les blocs algorithmiques et architecturaux qui composent le système HDR-ARTiSt. Le but est d'avoir tous les éléments permettant de capturer, calculer et afficher une vidéo HDR à une cadence de plusieurs images par seconde. Grâce à ces travaux, nous pouvons définir un système global, générique ne dépendant pas de l'architecture ciblée (FPGA, type de mémoire, capteur ou périphériques de communication).

Notre architecture n'est pas limitée à la seule résolution de notre capteur, ainsi qu'à un nombre défini d'images servant à la reconstruction de la carte de radiance. Le choix des algorithmes a été fait en s'inspirant à la fois de l'état de l'art des implémentations existantes, de tests qualitatifs des algorithmes au sein d'un environnement software spécialement conçu pour le HDR (*HDR-Lab*), ou encore en étudiant la complexité propre à chaque solution. Le système matériel ainsi conçu pourra être soumis aux tests en simulation hardware, ou directement embarqué sur la plate-forme, et c'est ce que nous allons voir dans le prochain chapitre.

La principale contribution à venir dans le Chapitre 5, sera la réalisation d'un prototype de test implémentant le système décrit dans ce présent chapitre. Nous présenterons les résultats d'implémentation, et des résultats visuels finaux directement observables en sortie du prototype.

# IMPLÉMENTATION ET RÉSULTATS DU SYSTÈME HDR-ARTIST

## 5.1/ INTRODUCTION

La totalité de notre architecture a été implémentée sous ISE/PlanAhead à partir de code en VHDL/Verilog. Du fait des contraintes imposées en début de thèse, aucun processeur n'a été utilisé, ni même de "black box" disponibles sous EDK. Seuls les utilitaires fournis par Xilinx avec la licence ISE, tels que les IPs du LogiCORE, les MIGs, les bibliothèques libres de droits ou les contrôleurs disponibles sur OpenCores ont contribué à l'élaboration de la solution. Les résultats d'implémentation pour deux images ( $P = 2$ ) et trois images ( $P = 3$ ) seront présentées dans ce chapitre. De même, deux résultats de synthèse et d'implémentation seront mis en avant, l'un à base du Virtex 5, et l'autre basé sur Virtex 6. L'implémentation sur Virtex 5 sera entièrement détaillée dans ce Chapitre. Pour comparaison, et pour éviter un trop grand nombre de données chiffrées, nous présenterons les résultats d'utilisation des ressources pour l'implémentation sur Virtex 6 seulement dans la Section 5.9.1. En faisant référence à une étude détaillée de la qualité visuelle de la première mise en œuvre, une amélioration à trois images appelée HDR-3 est décrite dans la section 5.7.

## 5.2/ ENVIRONNEMENT DE SIMULATION HARDWARE

Un des avantages de l'utilisation des langages VHDL et Verilog, est leur capacité à être utilisés à des fins de tests et d'essais, en stimulant le système par l'intermédiaire du contrôle des signaux. Afin d'accélérer la conception, ajuster ou tester une fonctionnalité du design, le développement d'un banc d'essai automatisé est nécessaire.

Il nous a fallu définir un environnement de simulation performant, capable de répondre à des exigences relatives aux conditions d'utilisation du matériel (synchronisation du capteur, bande passante mémoire, etc.). Nous avons eu besoin, au cours de cette thèse, de simuler tout d'abord la gestion de la mémoire. Pour une simulation cohérente et conforme, il a fallu nous positionner dans des conditions réelles de fonctionnement du système. Ces

contraintes doivent prendre en compte la cadence du capteur, ses périodes d'inter-trame et d'interligne, la bande passante de la mémoire, les vitesses de lecture/écriture dans la mémoire, les latences mémoires engendrées par la SDRAM, etc. Nous avons eu besoin, par la suite, de pouvoir simuler un résultat en sortie de notre système algorithmique, afin d'avoir un retour visuel de la création HDR et du tone mapping notamment. Un certain nombre de points cruciaux nécessaires au bon déroulement des simulations hardware sous Modelsim ont dû être respectés tels que :

- Simuler le capteur de façon entièrement contrôlée : génération des synchronisations, des pixels, de l'horloge pixel
- Pouvoir utiliser des fichiers en format image en entrée du système simulé en VHDL
- Pouvoir utiliser de vraies images issues du capteur en entrée de simulation
- Visualiser l'ensemble des signaux pour un débogage exhaustif

Afin de simuler notre capteur, nous nous sommes appuyés sur un article de Zuloaga et al. [124], aidant à réaliser un testbench relatif au traitement d'image. Ils proposent un pipeline complet de simulation. Dans un premier temps, une génération d'image est réalisée à partir d'un fichier texte de type PGM ("Portable GrayMap"), défini comme un simple tableau de pixels encodés au format hexadécimal. Ces pixels sont envoyés séquentiellement à un module VHDL qui les traite et les met en forme pour respecter les paramètres préalablement rentrés. Ces paramètres permettent de définir la fréquence pixel, les périodes d'interligne, d'inter-trame, et le lien vers le fichier PGM.

Les pixels sont ensuite envoyés vers les différents blocs de traitement. Une fois les traitements terminés, les pixels sont envoyés vers le bloc que nous appellerons MONITOR. Ce module se charge de convertir le flux de pixels arrivant en une image PGM directement exploitable sans conversion préalable. Ce bloc sert donc de jouer l'intermédiaire entre le flux de pixels et l'affichage, tel qu'on l'aurait sur un écran LCD classique.

Nous avons construit un pipeline plus complexe en utilisant le pipeline de simulation de Zuloaga. Le pipeline a été décliné en architecture parallèle synchronisée sur front d'horloge, envoyant  $P$  flux de pixel avec  $P$  fichiers texte en entrée, correspondant au  $P$  images LDRs. Le diagramme global de simulation hardware est disponible sur la Figure 5.1. Il en va de même pour la sortie et la visualisation des résultats écrits en parallèle dans plusieurs fichiers images.

### 5.3/ ESTIMATION DES TEMPS D'EXPOSITION AVEC LA MAE

Un problème s'est posé lorsqu'il a fallu décrire et simuler l'algorithme MAE en hardware de façon simple et efficace. La simulation n'a pas été faite entièrement, du fait de la trop grande complexité de l'environnement de simulation hardware à mettre en place. En effet, il aurait fallu fabriquer un environnement de simulation capable de simuler des changements de luminosité, pouvant gérer des flux vidéos entiers à partir de fichiers stockés sur



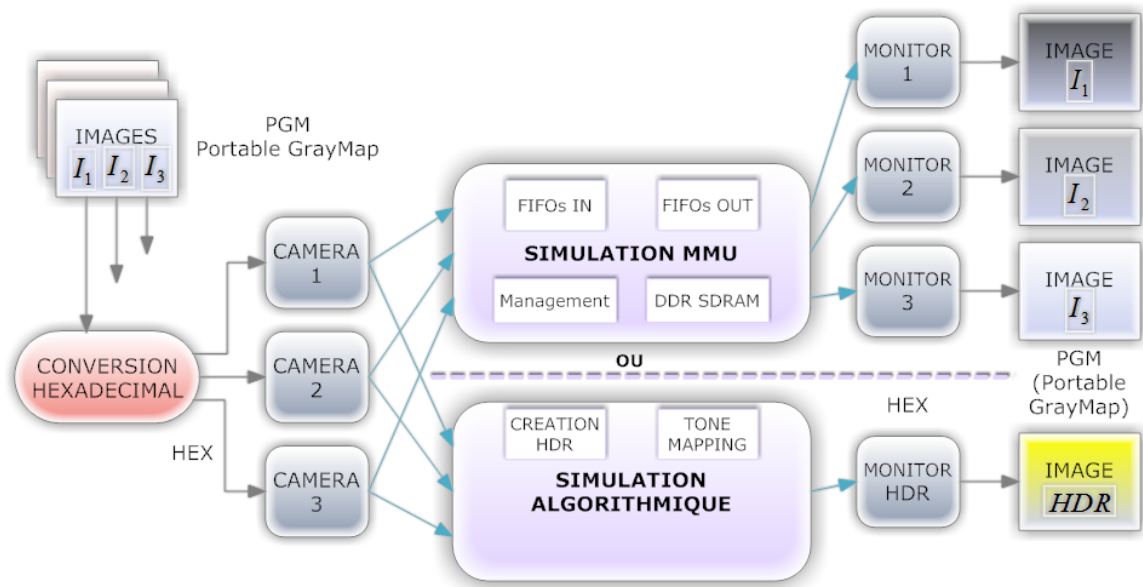


FIGURE 5.1 – Schéma du bloc de testbench hardware.

un PC, simuler la configuration, etc. La génération sous Modelsim d'une image à partir de fichier texte (voir la Section 5.2 précédente) prend beaucoup de temps. Il nous a été donc impossible de pouvoir simuler une vidéo complète avec un tel environnement. Le test de l'algorithme, notamment pour valider les temps de convergence des temps d'exposition, a directement été fait sur la plate-forme, de façon empirique, sans avoir fait de simulations au préalable. Néanmoins, d'autres aspects purement fonctionnels de l'algorithme MAE tels que le traitement des histogrammes (Eq.4.5), les sommes successives, les décisions (Eq.4.8,4.9,4.10), ont pu être simulés sous Modelsim.

## 5.4/ SIMULATION DE LA MMU

La plate-forme HDR-ARTiST, pour le prototype à Virtex 5, intègre une mémoire DDR2 de 256 MB avec un bus de data sur 64 bits (128 bits de bus en interface utilisateur), contre 512MB sur un bus de 64 bits (256 bits de bus en interface utilisateur) pour la plate-forme ml605 avec DDR3. Les opérations d'accès mémoire sont gérées par un contrôleur SDRAM personnalisé et spécifique à l'application. Nous utilisons la solution MIG ("Memory Interface Generator") de Xilinx, pour servir de point de départ au design du HDR-MMU ciblé pour la DDR SDRAM. Le générateur d'interface mémoire MIG est un outil utilisé pour générer des contrôleurs mémoires et des interfaces pour les FPGAs de chez Xilinx. La MIG génère les fichiers de conception Verilog ou VHDL, les contraintes UCF, les fichiers de simulation et les fichiers de script de mise en œuvre afin de simplifier et d'accélérer le processus de conception. L'utilisation du MIG a été fait à la fois pour la ml507, disposant d'un chip DDR2, et pour la ml605, dotée d'une DDR3. L'utilisation de



deux MIGs différents a malheureusement été inévitable. Du fait des différences fonctionnelles et descriptives entre les deux MIGs, la description complète a dû être refaite pour les deux architectures.

Un modèle de la SDRAM, sous forme de fichier, nous permet de simuler la réponse de la mémoire à nos commandes, sans disposer pour autant d'une vraie mémoire physique. Après avoir généré la MIG, un répertoire de simulation est créé, contenant un testbench, le modèle de la mémoire, et un fichier de simulation '.do'. Ces fichiers nous ont aidé à concevoir un testbench dédié à notre application, reprenant l'architecture de la Figure 5.1.

Une illustration de la machine à états (simplifiée) pour  $P = 3$  images est disponible en Annexe A.3. Elle sert à gérer les lectures et écritures des pixels en burst pendant les temps d'interligne du capteur. Elle est positionnée entre la MIG et les BRAMS.

La fréquence de fonctionnement de la mémoire SDRAM a été fixée à 200 MHz. Le capteur E2v capture 60 images par seconde avec un temps inter-trame de  $236\mu s$  et un temps de ligne de  $14,1\mu s$  ( $10\mu s$  pour les 1280 pixels et un temps entre les rangs de  $4,1\mu s$ ). Des BRAMS sont utilisées sur les entrées et sorties de la mémoire SDRAM, et agissent comme des tampons pour les données destinées à être lues ou bien écrites en rafale. Un ensemble de  $P$  différentes BRAMS est nécessaire : une BRAM utilisée pour alimenter la SDRAM avec les nouvelles données de capteur et un ensemble de  $P - 1$  BRAMS pour la lecture en flux parallèles, en sortie de la SDRAM, des trames capturées précédemment. Ces mémoires de blocs sont alimentées par des horloges indépendantes et ont donc des rapports non-symétriques pour les opérations d'entrées-sorties. La profondeur de chaque BRAM est de 2048 sur une largeur de bus de 10-bits, dans le but de pouvoir accueillir les 1280 pixels par rangée. Une ligne complète de 1280 pixels est écrite dans la mémoire en l'espace de  $\frac{1.280 \times 10}{128 \times 200 \cdot 10^6} = 0,5\mu s$ , où la vitesse d'écriture dans le bus de 128 bits est de  $200MHz$ . De même, le temps nécessaire pour la lecture d'une ligne de 1280 pixels de la SDRAM est identique. Ces deux opérations sont alors faiblement coûteuses en temps et sont effectuées au cours de l'intervalle de temps naturellement disponible entre les lignes (les synchronisations horizontales).

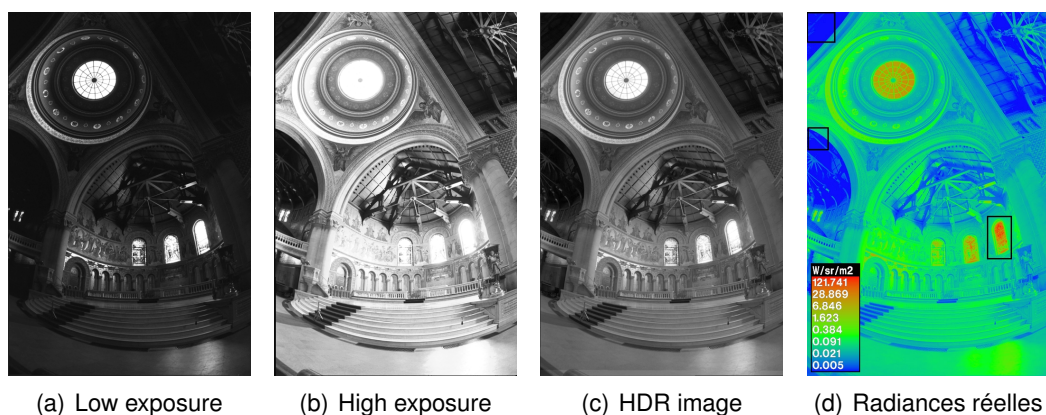
## 5.5/ SIMULATION DU SYSTÈME HDR-2

### 5.5.1/ DEBEVEC ET AL.

Comme nous l'avons vu dans la Section 4.5.2.4, la fonction de pondération de Debevec et al. [10] pose des problèmes notamment lorsque toutes les valeurs de pixel des images LDRs à une position  $i, j$  sont saturés blanc en même temps. Il en résulte des parties de l'image résultante grisées (voir la lampe de la Figure 4.15), alors que la valeur du pixel devrait être le maximum (à 255). Ceci est dû au fait qu'il peut y avoir une division

aberrante par 0 lorsque la pondération pour tout les pixels à une position  $i, j$  vaut 0, en éliminant totalement l'influence des pixels surexposés.

### 5.5.2/ LIMITES DU SYSTÈME HDR-2



**FIGURE 5.2** – Évaluation visuelle de l'application HDR-2 ( $P = 2$ ) en simulation hardware avec le mémorial de Paul Debevec. L'image fausse couleur (d) des valeurs de radiance réelle capturées avec 16 images LDRs dans la revue de Debevec [10], montre les zones de la Figure (c) où la quantité d'information est limitée.

Deux types d'évaluations, sur deux séquences d'images différentes ont servi à tester notre système.

Tout d'abord, la base de données classique mise à l'essai est une séquence de captures du mémorial de Stanford fournie par Debevec. Le système HDR-2 a été évalué sur l'image HDR issue de cette base de données Debevec (Figure 5.2). Il comporte une série d'images capturées avec des temps d'exposition allant de  $\frac{1}{1,024} s$  à  $32 s$ . Pour  $P = 2$ , un ensemble de deux images ont été choisi afin de simuler l'architecture matérielle complète. La Figure 5.2 représente un exemple d'image HDR reconstruite par l'architecture HDR-2 avec une exposition faible ( $\Delta t_L = 0.5 s$ ) et une exposition élevée ( $\Delta t_H = 16 s$ ). Il convient de noter que les expositions sont relativement élevées, en raison des conditions d'éclairage à l'intérieur de l'église. L'image tone mappée reproduit les détails dans les zones sombres et claires sans artefacts visuels perceptibles. À titre comparatif, nous avons ajouté la carte de radiance fausse couleur 5.2(d) issue de la revue de Debevec [10]. On remarque que certaines zones de l'image résultante du HDR à deux images manquent de détails. Cette remarque est à contre balancer, car la majorité des zones sont malgré tout bien représentées.

Deuxièmement, nous avons aussi évalué le système HDR-2 avec des captures du monde réel, réalisées par la plate-forme HDR-ARTiSt (Figure 5.3 (a)). La scène expérimentale est un bureau sur lequel on peut trouver une tasse de café et quelques jouets (une voiture, une figurine Toy Story, un robot R2D2 à l'intérieur de la tasse de café). Une lampe a été

placée derrière cette scène afin d'augmenter considérablement la plage dynamique si celle-ci est allumée (voir Figure 5.3 (c) et la Figure 5.3 (d)).

Nous calculerons ici la dynamique de la scène capturée de la Figure 5.3 grâce à la caméra HDR-ARtiSt :

1. Nous cherchons tout d'abord à trouver une image pour laquelle il n'y a pas de pixels saturés haut. Nous faisons varier le temps d'exposition par l'intermédiaire des registres du capteur, à ouverture fixe, pour avoir une image 8 bits où la valeur de pixel maximale vaut 254 (donc pas de saturation blanche). Nous trouvons alors  $\Delta t_L = 15.72\mu s$

2. On applique la formule d'irradiance de Debevec suivante :

$$\ln E_{ij,max} = g(Z_{ij,254}) - \ln \Delta t_L \quad (5.1)$$

3. On trouve alors  $E_{ij,max} = 258998 W.sr^{-1}.m^{-2}$ , avec  $g(Z_{ij,254}) = 1.404$
4. On effectue une manipulation similaire pour trouver une image avec très peu de pixels saturés noir, et on a alors :  $\Delta t_H = 13.2830ms$
5. On applique la formule et on obtient  $E_{ij,min} = 2.34.sr^{-1}.m^{-2}$ , avec  $g(Z_{ij,254}) = -3.470$
6.  $E_{ij,min} = 2.34 W.sr^{-1}.m^{-2}$
7. On calcul alors la dynamique à partir de l'équation 2.4. Et on trouve  $D_y = 102.2dB$ .

Avec une telle dynamique, on peut rapprocher notre scène avec la dynamique de la scène de Debevec qui est d'à peu près  $107.9dB$ . Pour ce qui est de la dynamique théorique captable par notre caméra, elle dépend de la plage de temps d'exposition configurable sur celle-ci. Pour le capteur e2v, le temps d'exposition minimal est théoriquement de  $\Delta t_{min} = 15.72\mu s$  et le temps d'exposition maximal configurable est de  $\Delta t_{max1} = 1s$ . Cette plage correspondrait à une dynamique de  $D_{y,max} = 108.27dB$ . En mode vidéo, et à pleine résolution, le temps d'exposition maximal n'est malheureusement pas cette valeur, puisque il s'y ajoute une contrainte temporelle pour respecter la grande cadence de  $60fps$  du capteur. L'exposition maximale vaut plutôt  $\Delta t_{max2} = 16.09ms$ , ce qui correspond à une dynamique maximum de  $D_{y,max2} = 102.54dB$ .

De toute évidence, la principale difficulté pour la technique HDR est à la fois de faire apparaître les détails dans la zone lumineuse proche de l'ampoule, et dans la zone sombre à l'intérieur de la tasse de café ; tout en préservant un bon contraste pour les niveaux intermédiaires. Notez que les expositions sont ici très faibles (respectivement  $0,15ms$  et  $1,9ms$  pour l'exposition faible et élevée) en raison de l'"aveuglement" créé par la source de lumière.

Une première analyse visuelle montre des résultats similaires à ceux obtenus précédemment avec l'église, donnant une reproduction de haute qualité et des détails fins dans les zones sombres et éclairées. À titre d'exemple, la figure R2D2 peut même être détectée

à l'intérieur de la tasse où il fait particulièrement sombre. De l'autre côté, le mot "HDR" écrit dans l'abat-jour peut être lu<sup>1</sup>. Cependant, une analyse plus approfondie révèle un manque de détails dans les zones de luminosité moyenne, mettant en évidence certaines limites du système HDR-2. Quatre imageries différentes ont été choisies pour souligner les artefacts possibles. Ces zones sont agrandies et affichées dans la partie inférieure de la Figure 5.3 (b), révélant une certaine discontinuité le long des bords de la tasse de café (zone 1), sur le capot de la voiture (zone 2), à l'ombre de la voiture (zone 3), et dans l'abat-jour (zone 4). Ces phénomènes observables sont principalement dus à la grande gamme dynamique de la scène (environ 250000 : 1), et du fait d'expositions trop écartées. Dans de telles conditions d'éclairage, il semble presque impossible à capter simultanément les informations à la fois sur l'ampoule, sur la partie sombre de la scène, et sur toute la plage intermédiaire restante avec seulement deux images. Pratiquement, pour plus de détails à la capture de la partie supérieure de l'histogramme (lampe), le temps d'exposition de la première image doit être diminué fortement. De l'autre côté, pour les zones très sombres, le temps d'exposition de la seconde image est augmenté considérablement. Ainsi, avec seulement deux captures, il n'est pas possible de capturer la scène dans tout ses détails. Ceci implique inévitablement une absence d'informations dans la zone médiane de l'histogramme, ce qui conduit aux artefacts mentionnés ci-dessus.

## 5.6/ IMPLÉMENTATION COMPLÈTE DE HDR-2

### 5.6.1/ RÉSULTATS DE SYNTHÈSE MATÉRIELLE SUR VIRTEX 5

Le Tableau 5.1 résume toute l'architecture et le Tableau 5.2 décrit en détail le résultat de synthèse sous ISE. Habituellement, les traitements d'image à base de FPGA nécessitent de nombreux appareils spécifiques tels que la mémoire externe SRAM, une mémoire multi-port, un DMA ("Direct Memory Access"), des processeurs dédiés, qui consomment beaucoup de blocs DSPs. Ce n'est pas totalement le cas pour notre système. On peut voir dans le Tableau 5.2 que les résultats de synthèse ont une complexité matérielle relativement faible puisque le nombre de LUTs est de 12859 (environ 29% des ressources du FPGA) et le nombre de flip-flops est 13612 (soit 30% des ressources disponibles) pour la configuration HDR-2.

Une partie importante de la complexité est due à la gestion de mémoire MMU-P qui gère les synchronisations des flux de pixels entre la mémoire externe SDRAM et les BRAMS en interne (environ 5% des LUTs et 6% de flip flops). Seulement 15,5% de FIFO/BRAMS sont empruntées : les BRAMS sont principalement consacrées au stockage des  $P$  lignes de pixels nécessaires aux processus de création HDR. Les FIFOs sont dédiées à la synchronisation des données entre les différentes étapes du pipeline HDR. La section de

---

1. La figure R2D2 et le mot "HDR" ne sont visibles qu'en version électronique de ce document et sont presque cachés dans la version imprimée, à cause de la gamme dynamique limitée des imprimantes.

Clock Domains			
Interface capteur	contrôleur SDRAM	Traitement HDR	Contrôleur DVI
114Mhz	200MHz	114Mhz	25Mhz

Paramètres du système			
P	Résolution	Sortie	FPS
2	1,280 × 1,024	78.6 Mpixels/sec	60

**TABLE 5.1** – Résumé des caractéristiques communes aux deux prototypages sur ml507 et ml605 (Virtex 5/6).

création HDR et les processus de tone mapping consomment respectivement 9,4% et 7,7% des LUTs et 11,4% et 9% de flip flops en raison de la mise en œuvre d'opérateurs arithmétiques IEEE754 32-bit à virgule flottante (en particulier les diviseurs et les multiplieurs).

Théoriquement, avec un nombre de slices d'environ 6000, notre architecture serait capable de tenir sur un FPGA plus petit, de type Spartan 6 LX45, doté de 6822 slices et qui coûte entre 50 et 80 euros.

	Utilisé	Disponible	Utilisation (%)
<b>Logique</b>			
Nombre de Slices occupés	5,272	11,200	47.1%
<b>Distribution de la complexité</b>			
Slice LUTs	12,859	44,800	28.7%
MMU	2,149		4.8%
Traitement HDR	4,211		9.4%
Tone mapping	3,433		7.7%
Post processing	3,066		6.8%
Slice Flip Flops	13,612	44,800	30.4%
MMU	2,682		6%
Traitement HDR	5,107		11.4%
Tone mapping	4,045		9%
Post processing	1,778		4%
FIFO/BRAM	23	148	15.5%

**TABLE 5.2** – HDR-2 : Résumé du rapport d'implémentation matériel sur Virtex 5

L'étape de post-traitement est également une tâche consommatrice de ressources (7% des LUTs et 4% de flip flops). Cette tâche embarque un contrôleur DVI dédié, conçu pour afficher les données de la caméra sur un écran LCD. Les deux flux vidéos en streaming, le flux de pixels LDRs non transformés ou après traitement (HDR+tone mapping),

peuvent être affichés sans latence perceptible pour le spectateur. À cet effet, les signaux de synchronisation horizontale et verticale ( $H_{sync}$  et  $V_{sync}$  dans la Figure 4.6) sont régénérées à partir de la sortie post-traitement pour synchroniser le décodeur DVI avec les pixels traités ou non. Même si ce contrôleur DVI consomme assez peu de ressources, il ne peut pas être considéré comme une partie intégrante de l'application. Il est uniquement utilisé pour visualiser les données de sortie sur le moniteur LCD, où les vidéos des différentes étapes du pipeline (trames LDRs et trames HDRs après traitement) peuvent être inspectées indépendamment. Ainsi, en ignorant les ressources du contrôleur DVI, notre mise en œuvre de l'application HDR sur la plate-forme HDR-ARTiSt a une complexité matérielle relativement faible. Cela ouvre des possibilités intéressantes pour d'autres applications de traitement d'images sur FPGA.

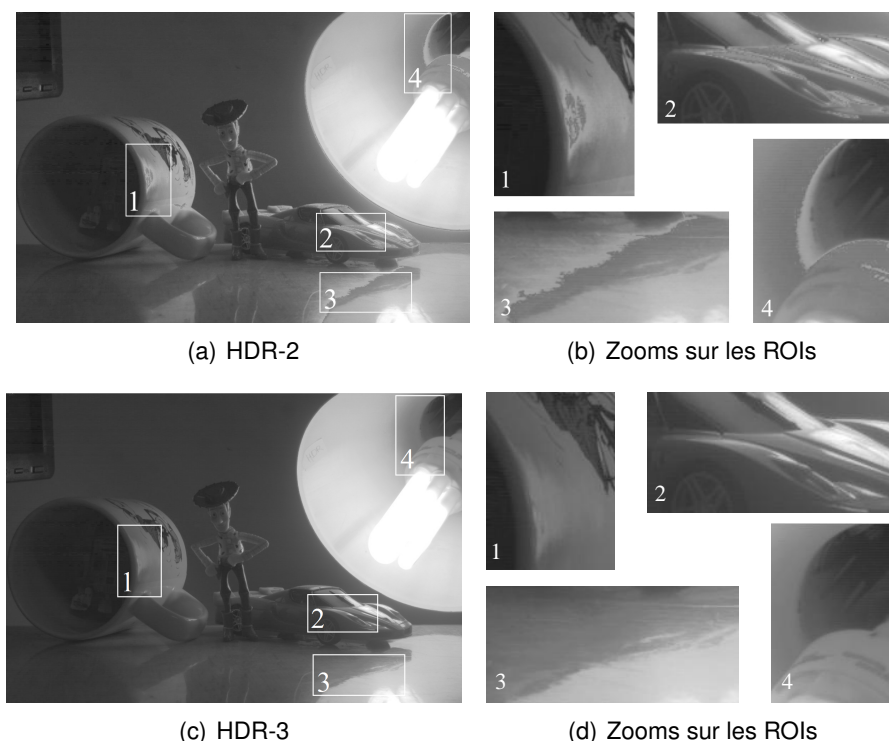
### 5.6.2/ LATENCE OBSERVABLE

Notre capteur a un temps d'interligne appelé  $h\_sync$  ("h" pour horizontale), de 307 pixels, et un temps d'inter-trame  $v\_sync$  ("v" pour verticale) de 20 lignes. Le système algorithmique contient une latence fixe que nous pouvons calculer. Le temps de latence à la fin de chaque ligne, après traitement de cette même ligne, est de 127 coups d'horloge (soit  $1,11\mu s$  pour une horloge pixel cadencée à  $114MHz$ ). Cette latence constante apparaît mais ne modifie pas la fréquence de trame  $fps$ . En effet, nous utilisons les temps d'interligne et d'inter-trame livrés par notre capteur pour combler et compenser cette latence. Avec une fréquence vidéo de 60 images par seconde, notre système est capable de traiter  $60 \times (1280 + 307) \times (1024 + 20) = 99,40$  Megapixels par seconde. Notre système matériel a une fréquence maximale d'exploitation de  $126,733MHz$ , de telle sorte que notre système est capable de traiter chaque pixel à chaque coup d'horloge. Enfin, notre architecture intègre tous les opérateurs algorithmiques pour produire un pixel de sortie HDR ou tone mappé en temps réel.

## 5.7/ IMPLÉMENTATION DE HDR-3

Pour contourner la limite de fonctionnement décelée avec l'application HDR-2, la solution la plus naturelle consiste à augmenter le nombre d'images pour la reconstruction de l'image HDR. Avec la complémentarité apportée par une exposition intermédiaire, HDR-3 est capable de capturer les données manquantes dans ces conditions lumineuses. L'image de la scène obtenue expérimentalement en utilisant cette nouvelle implémentation est représentée sur la Figure 5.3 (c). On voit que les divers objets présents dans la Figure 5.3 (d) sont maintenant visibles et dépourvus d'artefacts dans cette nouvelle image.

La Figure 5.4 présente une image récupérée en sortie de la caméra, et une image gé-



**FIGURE 5.3** – Comparaison entre le système HDR-2 et HDR-3. On y aperçoit les améliorations apportées par le système vidéo HDR-3 dans des conditions d'éclairage extrêmes.

née de façon software. Toutes les deux sont générées à partir de la même séquence d'images LDRs  $I_L$ ,  $I_M$  et  $I_H$ , de la même courbe de réponse inverse  $g$ , et de la même fonction de pondération  $w$ . On peut remarquer qu'il y a très peu de différence entre les deux images, la version hardware donnant généralement plus de détails dans les zones éclairées, tandis que la version software donne une légère différence de contraste au niveau des basses luminosités (voir intérieur du tube). Ceci est dû au fait que le mode de représentation (IEEE754), la dynamique de représentation et l'erreur de quantification des données flottantes sont identiques entre la version software (Matlab, C++) et la version hardware (VHDL). Il n'y aurait aucune raison justifiant un résultat différent, du moment que l'on utilise la même méthode IEEE754 en 32 bits pour les deux descriptions.

Dans le Tableau 5.3, un résumé de la complexité matérielle des différents blocs est donné. La synthèse de HDR-3 se traduit évidemment par une complexité légèrement plus élevée que celle de HDR-2 (+11,7%). Parmi les sous-modules figurant dans le Tableau 5.3, la quantité de ressources logiques consommées par la MMU (+31,6% de LUTs) et la création HDR (+35,9% de tables LUTs et +11,2% de bascules) sont augmentées. De plus, lorsque le nombre d'images passe de deux à trois, une augmentation significative de l'utilisation de blocs RAM (+30,4%) est remarquable, principalement dû à la gestion de la mémoire MMU. Contrairement à ces augmentations importantes, le module de tone mapping ne subit aucun changement et consomme les mêmes ressources. Les autres modules du système sont relativement moins touchés, à l'exception du nombre de bas-



(a) Set.  $\Delta t_L = 0.094ms$ ,  $\Delta t_M = 1,26ms$  et  $\Delta t_H = 5,36ms$ 

(b) Version Software



(c) Version Hardware

**FIGURE 5.4** – Comparaison entre les versions software et hardware du pipeline HDR-3 (Debevec+Duan).

cules pour le sous-module de post-traitement. En effet, ce module intègre des ressources supplémentaires (registres, multiplexeurs, etc.) nécessaires pour gérer les trois flux LDRs et le flux HDR à acheminer vers la sortie DVI.

## 5.8/ EXTENSION PAR LA COULEUR

La plate-forme HDR-ARTiSt existante N&B peut facilement être améliorée avec un capteur couleur. Dans notre cas, les cartes capteur construites sont compatibles pin à pin avec la version du capteur couleur d'e2v. Dans une génération conventionnelle de contenu HDR, les étapes de rendu et de génération de la carte de radiance sont appliquées après dématricage. Toutefois, comme expliqué dans [125], le système visuel humain (SVH) effectue la plupart des traitements d'adaptation sur la matrice des cônes avant dématricage. Selon ce principe, Tamburrino et al. [125] propose un nouveau fonctionnement où le tone mapping est fait directement sur la matrice issu du filtre couleur (CFA) de l'image. Le dématricage est alors la dernière étape du pipeline HDR. Une telle approche est alors directement compatible avec le pipeline HDR mis en œuvre sur notre plate-forme HDR-ARTiSt. En effet, les sections de création HDR et de tone mapping peuvent être appliquées sur les pixels de l'image mosaïquée à l'aide de trois courbes de réponse différentes du capteur (pour les trois composantes RVB), avant d'appliquer le dématricage. En termes de calcul, il ne s'agit pas d'augmenter significativement la complexité des algorithmes, car il n'y a



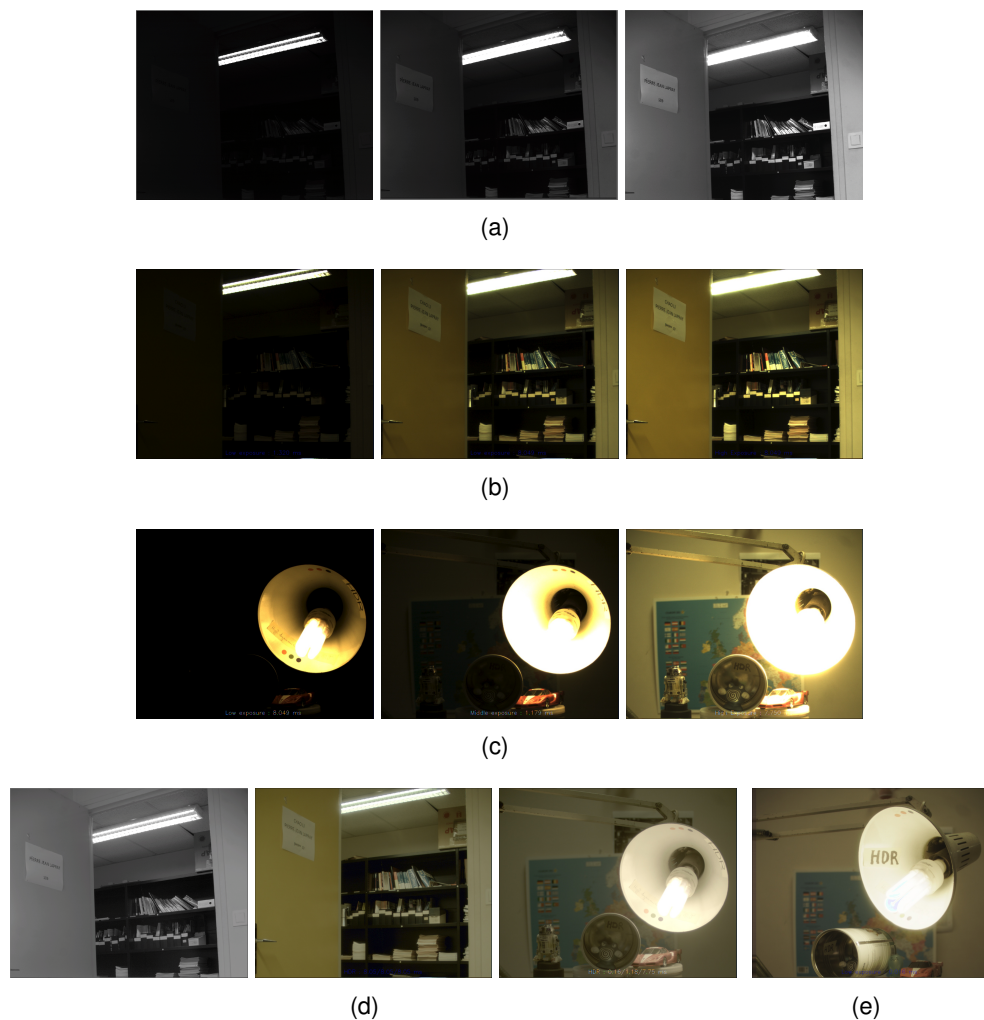
	Utilisé	Utilisation (%)	Variation avec HDR-2
<b>Utilisation Logique</b>			
Slices Occupés	5,891	52.6%	+11.7%
<b>Distribution de la complexité</b>			
Slice LUTs	15,281	34.11%	+18.8%
MMU	2,829	6.3%	+31.6%
Traitement HDR	5,722	12.8%	+35.9%
Tone mapping	3,433	7.7%	0%
Post processing	3,297	7.4%	+7.5%
Slice Flip Flops	15,134	33.8%	+11.2%
MMU	2,682	6%	+0.1%
Traitement HDR	6,071	13.5%	+18.9%
Tone mapping	4,045	9%	0%
Post processing	2,333	5.2%	+31.2%
FIFO/BRAMs	30	20.3%	+30.4%

**TABLE 5.3** – HDR-3 : Résumé du rapport de synthèse hardware sur Virtex 5.

qu'un seul flux de données RAW à traiter et non trois flux de composantes différentes. L'étape de dématricage, qui convertit les données RAW en données RVB, est géré par un bloc matériel spécifique (appelé "interpolation du filtre couleur") et est fourni par Xilinx. Ce bloc génère les composantes de couleurs manquantes associées à la matrice de Bayer utilisée dans les systèmes photographiques. La Figure 5.5 montre des images issues du passage du système noir et blanc à un système couleur.

En ce qui concerne ressources matérielles, seulement deux LUTs supplémentaires de taille  $1024 \times 32$  sont nécessaires afin de stocker les courbes de réponse, et un opérateur de dématricage. Selon Debevec et al. [10], on peut aussi, pour plus de simplicité, utiliser une courbe identique et unique pour les composantes R et G parce que ces deux courbes de réponse sont très proches.

Pour ce qui est de la qualité du résultat obtenu, on remarque que toutes les informations et les contrastes dans les images en noir et blanc sont bien présents dans l'image résultante. On note néanmoins que les couleurs sont globalement délavées. Des améliorations sont à apporter vis à vis de cet aspect, en implémentant, par exemple, un module de "white balancing", ou un tone mapping adéquat. À ce propos nous avons aussi implémenté l'algorithme de tone mapping de Reinhard global, ayant un rendu des couleurs moins délavé mais ayant des problèmes chromatiques dans les hautes luminosités (voir Figure 5.5(e)).



**FIGURE 5.5** – Images N & B (a) et couleurs (b) et (c) issues du capteur couleur de chez E2v, et les résultats HDRs obtenus avec le tone mapping de Duan et al. (d) et Reinhard et al. (e).

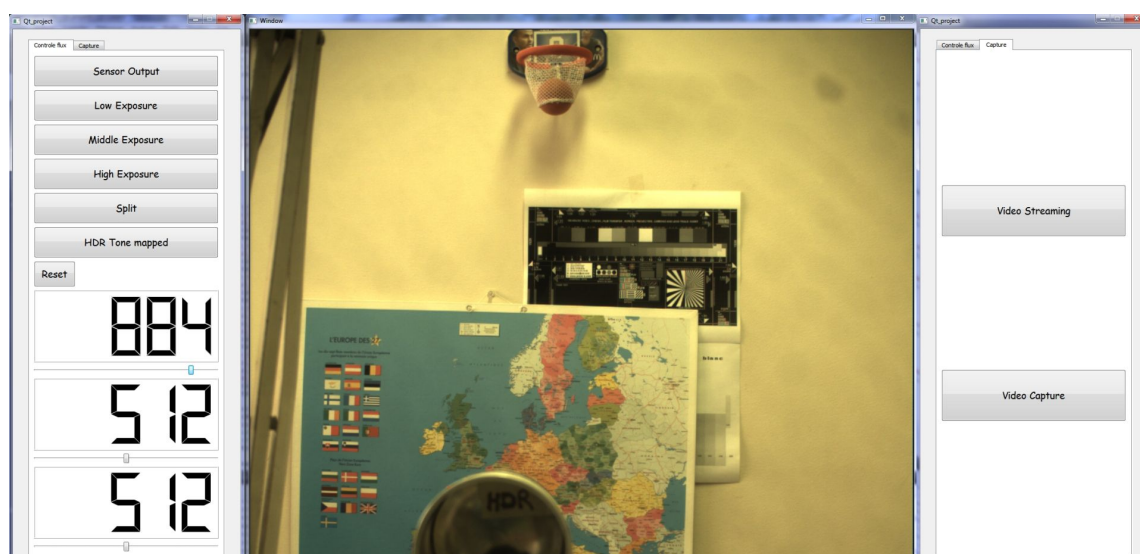
## 5.9/ EXPÉRIMENTATIONS ET CONCLUSIONS

### 5.9.1/ IMPLÉMENTATION SUR VIRTEX 6

Nous présenterons ici les résultats d'utilisation des ressources pour une implémentation HDR-2 et HDR-3 sur la plate-forme munie d'un Virtex 6 (voir Figure 5.4). Les résultats sont différemment présentés par rapport aux implémentations sur Virtex 5 précédentes. Nous présentons plutôt les utilisations par composants FPGA et non par composants (modules) du design. Du fait de la très grande superficie du Virtex 6, on remarque que l'utilisation relative est très faible, ceci encourage une implémentation sur un FPGA beaucoup plus petit et donc à coût très réduit.

**TABLE 5.4** – Résumé de l'implémentation HDR-2 et HDR-3 pour la plate-forme Virtex 6.

Métrique	HDR-2	HDR-3	Diff. (%)
Puissance	6.019 W	6.039 W	0%
Fréquence maximum	139.6 MHz	125.0 MHz	-10%
Slices empruntés	4,897 out of 37,680 (12%)	6,692 (17%)	+5%
LUTs	12,630 out of 150,720 (8%)	16,880 (11%)	+3%
Registres	14,916 out of 301,440 (4%)	20,192 (6%)	+1%
Nombre d'IOBs	195 out of 600 (32%)	196 (32%)	
BRAMs 36K	7 out of 416 (1%)	17 (4%)	+3%

**FIGURE 5.6** – Aperçu de l'application HDR-ViewFinder réalisée sous Qt, permettant l'interaction avec la caméra HDR-ArtiSt via le protocole UDP par liaison Ethernet.

### 5.9.2/ HDR-Viewfinder : APPLICATION ETHERNET AVEC QT

Une application nommée *HDR-Viewfinder* a été conçue sous Qt. Elle gère le protocole Ethernet dans le but de pouvoir récupérer des échantillons vidéos et interagir avec la caméra HDR-ArtiSt (voir Figure 5.6).

En ce qui concerne le protocole Ethernet, nous avons implémenté sur le FPGA le module "Embedded Tri-Mode Ethernet MAC" fourni par Xilinx, en le faisant interagir avec un module de gestion du protocole UDP trouvé sur le site OpenCore. Nous ne détaillerons pas cette implémentation. Les fonctionnalités de l'application Qt sont les suivantes :

- Sélectionner et afficher les flux de la caméra (voir les différentes sorties disponibles en dans la Figure D.1 Annexe D)
- Configurer manuellement jusqu'à trois temps d'exposition dans le cas où la MAE (Multiple Auto Exposition) est désactivée

- Faire un reset des synchronisations et du capteur
- Visualiser en direct la sortie vidéo ("video streaming")
- Capturer une séquence vidéo sous plusieurs formats, tout en gardant le contrôle de la caméra (sélection de la sortie, changement des registres, etc.)

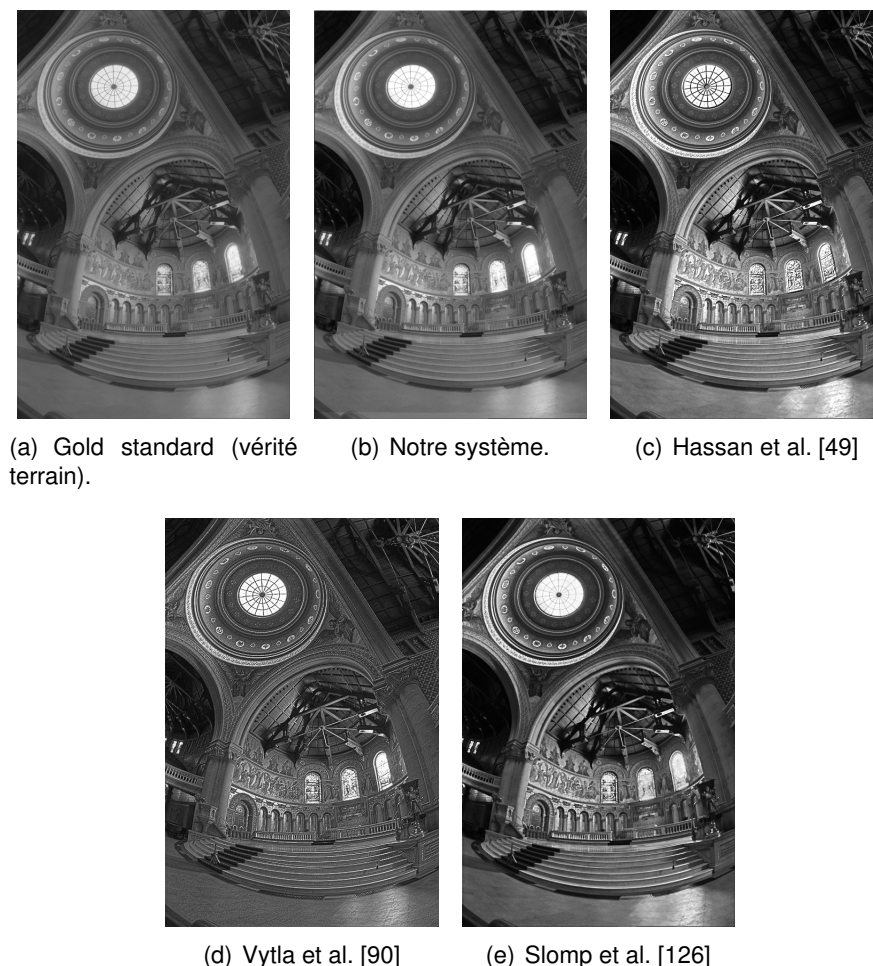
Une autre application a été réalisée via un protocole RS232, implanté dans le FPGA pour configurer un à un les registres du capteur depuis le PC. Nous ne détaillerons pas cette application.

## 5.10/ COMPARAISON AVEC LES ARCHITECTURES EXISTANTES

En plus des résultats de performance, une comparaison de HDR-3 avec trois autres systèmes a été faite. Deux d'entre eux sont sur une architecture à base de FPGA [49, 90] et le dernier à base de GPU [126]. Cependant, ces méthodes de tone mapping utilisent une image commune en entrée de leur système. En effet, les architectures de vision matérielle pour la capture et le rendu HDR en vidéo live sont à un stade prématuré [127], et nous n'avons pas trouvé de système HDR complet dans la littérature. L'architecture HDR-3, d'autre part, est un prototype de caméra innovant mettant en œuvre un pipeline HDR complet. Du point de vue purement algorithmique, notre puissance de calcul en temps réel dépasse les capacités de calcul des trois autres implémentations.

Le Tableau 5.5 résume les résultats de comparaisons en termes de performances de traitement (fps, résolution). En performances brutes, notre architecture fonctionne à 60 images par seconde sur une résolution de  $1.280 \times 1.024$  pixels, donnant un rendu global de 78,6 millions de pixels par seconde. Cette performance est significativement plus élevée que les deux autres architectures à base de FPGA et inférieure à la mise en œuvre sur GPU. Néanmoins, la carte Nvidia 8800GTX utilisée dans cette variante GPU n'est pas bien adaptée à une faible consommation d'énergie typique des systèmes embarqués et caméras intelligentes.

Nous avons également comparé la qualité des images produites par les différentes architectures en utilisant cinq indicateurs : l'erreur quadratique moyenne (MSE), la racine de l'erreur quadratique moyenne normalisée (NRMSE), le ratio Peak Signal-to-Noise (PSNR), l'indice de qualité universelle (UQI) [107], et la similarité structurelle (SSIM) [128]. Le facteur UQI correspond à peu près à la perception humaine que l'on aurait afin de juger les différences entre les 2 images. La valeur de UQI entre les images est comprise dans l'intervalle  $[-1, 1]$ , et est égale à 1 pour des images identiques, 0 si les images ne sont pas du tout corrélées, et -1 si les images sont anticorrélées. La similarité structurelle (SSIM) est une méthode pour mesurer la similarité entre deux images. L'indice SSIM peut être considéré comme une mesure de qualité, à condition que l'autre image soit considérée comme étant d'une qualité parfaite. Il s'agit d'une version améliorée de l'indice UQI.



**FIGURE 5.7** – Comparaison de la sortie du système proposé avec les différentes méthodes de tone mapping en temps-réel existantes.

Notre image "d'or" est l'image de l'église de Stanford tone mappée avec la méthode de Drago [13], à partir d'une carte de radiance construite à partir de 16 images différentes. Cette technique est décrite comme la plus naturelle qui existe, et aussi comme la méthode la plus performante pour obtenir un bon rendu dans les régions sombres d'une scène [63]. Considérant cette image comme la référence source parfaite, le MSE, le PSNR et le UQI ont été calculés pour chaque architecture.

L'ensemble des images sont visibles sur la Figure 5.7. La méthode de Hassan [49] semble apporter plus de contraste et de détails dans l'image que notre méthode. Ceci est principalement dû au fait qu'ils utilisent en entrée de test des images HDRs construites à partir d'une carte de radiance idéale, obtenue "hors ligne", à partir d'un ensemble de 16 expositions multiples de la même scène. Elle est issue du site web de Paul Debevec. Cependant, l'évaluation visuelle des différentes images révèle que le système HDR-3 donne des résultats comparables à l'étalon-or. L'architecture HDR-3 a la plus faible erreur quadratique moyenne, la plus faible NRMSE et le plus haut PSNR. En outre, avec une valeur UQI de 0,90 et un SSIM de 0,93, cela signifie que les images HDRs fournies

	This work	FPGA [49]	FPGA [90]	GPU [126]
Device	Virtex 5/6	Stratix II EP2S130F780	Stratix II EP2S15F484	Nvidia Ge- Force 8800 GTX
Frame rate (fps)	60	60	60	100
Frame size (pixels)	1,280 × 1,024	1,024 × 768	1,024 × 1,024	1,024 × 1,024
MSE	286.95	668.01	386.46	586.07
NRMSE	0.17	0.38	0.13	0.19
PSNR (dB)	23.55	19.88	22.26	20.45
SSIM	0.93	0.69	0.55	0.70
UQI	0.90	0.71	0.67	0.70

**TABLE 5.5** – Comparaisons avec les architectures existantes de tone mapping en temps réel.

par notre architecture sont les plus proches de l'étalon-or. Le rendu visuel peut donc être considéré comme le plus naturel si l'on se réfère aux études démontrant la qualité de cet algorithme.

### 5.11/ PERSPECTIVE : LE PROBLÈME DE GHOST

Un des problèmes majeurs et bien connu de l'imagerie HDR par multiple expositions, en plus du cas de la "colour constancy", est la difficulté de supprimer les effets indésirables de mouvement apparaissant lors de la reconstruction de la carte de radiance. L'analyse du mouvement et la correction des artefacts sont des voies de recherche afin de fournir une meilleure qualité de vidéo HDR. Le système actuel de bracketing est limité par les dissemblances spatio-temporelles qui peuvent arriver pendant la capture des images. Ces artefacts, nommés "ghost", peuvent être de deux natures : globaux ou locaux. Les ghost globaux proviennent d'images LDRs mal alignées, surgissant lors du mouvement de la caméra elle-même, lorsque l'on filme avec une caméra tenue à la main par exemple (sans stabilisateur). L'autre type de ghost provient de mouvement d'un objet ou d'une personne dans la scène pendant l'acquisition. Cette anomalie peut rendre inexploitable l'imagerie HDR dans certains domaines d'applications.

Dans notre cas, nous n'avons pas implémenté de correction de mouvement, ce qui peut poser des problèmes lorsqu'il y a présence d'une assez grande quantité de mouvement dans la scène. Un exemple de ces effets indésirables apparaissant sur notre image HDR réalisée à partir de trois images est montré dans la Figure 5.8. On remarque que le ghost engendré par le mouvement du doigt peut présenter plusieurs intensités. Le ghost sera faible par exemple si une des images ayant servi à construire l'image HDR (en l'occurrence l'exposition faible pour la Figure 5.8(a)), où les pixels que composent le

doigt sont tellement sombres qu'ils ne sont pas pris en compte lorsque la fonction de pondération est appliquée. A l'inverse, si sur les trois images le doigt est bien visible, et bien contrasté avec le fond, alors l'intensité du ghost sera grande (voir Figure 5.8).

Nous n'avons pas trouvé d'architectures comparables implémentant ce genre d'algorithme sur plate-forme matérielle et en temps-réel. Toutefois, une "smart camera" de la littérature [129], est capable de calculer dans le flux de pixel une image-différence seuillée à partir de deux images, de détecter les coordonnées des pics de différences après projections horizontales et verticales, et ainsi de repérer le mouvement dans les images. Une approche similaire pourrait être envisageable pour évaluer la dynamique de mouvement entre les trois images LDRs.

On prend les trois images  $I_L$  (basse),  $I_M$  (moyenne) et  $I_H$  (haute) et on cherche le mouvement entre la  $I_L/I_M$  et  $I_M/I_H$  :

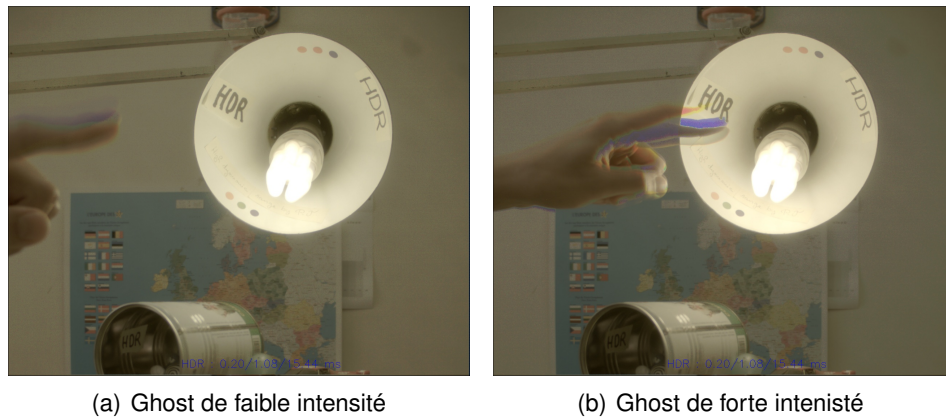
1. On "booste" l'image  $I_L$  pour faire correspondre le mieux possible ses intensités de pixels à celle de l'image  $I_M$
2. On enregistre une cartographie des facteurs de boost qui nous ont permit de faire correspondre les images
3. On calcule l'image-différence entre  $I_M$  et  $I_L$  (boostée) et on binarise la différence après seuillage. On obtient alors une image binaire, avec les coordonnées de l'image où apparaît le mouvement
4. On fait la même chose pour  $I_M/I_H$  en parallèle au sein du FPGA
5. Pendant la triple capture suivante, on aura les coordonnées du mouvement pour  $I_L/I_M$  et  $I_M/I_H$
6. Pour supprimer les ghosts, on affecte les contours de l'image  $I_M$  sur  $I_L$  et  $I_H$ , aux coordonnées où on aura effectivement repairé du mouvement (en prenant en compte les facteurs de boost calculés précédemment).

Le but serait de reconstruire l'image HDR toujours avec les trois images LDRs, mais avec les expositions extrêmes corrigées avec les contours présents dans l'image  $I_M$ .

## 5.12/ CONCLUSION

Le système HDR-ARTiSt a été testé et implémenté sur deux plates-formes différentes. Les résultats d'utilisation des ressources sont assez encourageants, dans le sens où il est possible de migrer l'architecture vers du "low cost". Nous avons aussi amélioré la caméra en lui implantant un capteur couleur, augmentant par là même la quantité d'information perçue (information couleur). Un logiciel HDR-ViewFinder a été conçu pour piloter la caméra via Ethernet, pouvant modifier les temps d'exposition, choisir le flux de





**FIGURE 5.8** – Ghost apparaissant au sein du système HDR-ARTiSt.

sortie, enregistrer images et vidéos, etc. Le pilotage de la caméra se fait sans reconfiguration préalable du FPGA. On peut aussi imaginer de faire un design hardware de caméra se passant des cartes d'évaluation de Xilinx pour réaliser un système portable. Pour le moment le test de la caméra est limité dans un environnement restreint, puisque nous n'avons pas la possibilité de transporter la caméra et de récupérer des vidéos via Ethernet en dehors du bureau de test.





## CONCLUSION

### 6.1/ PRINCIPALES CONTRIBUTIONS

Dans cette thèse, nous présentons un prototype matériel complet de vision appelé HDR-ARtiSt, basé sur un capteur d'image standard associé à une carte d'évaluation FPGA. Cette caméra intelligente dédiée à la vidéo "live" remplit la contrainte de temps réel que nous nous sommes fixée, tout en satisfaisant les exigences de qualité d'image. Il embarque un pipeline HDR complet, sachant faire des captures successives de plusieurs images, réalise la création de la carte de radiance de la scène, le tone mapping, et ressort un streaming vidéo HDR sur un écran LCD.

Le système a été construit comme une plate-forme adaptative. Du point de vue purement matériel, notre pipeline HDR peut être facilement adapté à de nombreux capteurs CMOS conventionnels. L'utilisation d'un nouveau capteur ne nécessite que la conception d'une carte-capteur spécifique dans le but d'être branchée sur la carte mère et reliée au FPGA. L'évaluation d'une nouvelle fonction de transfert de la caméra est néanmoins nécessaire. D'un point de vue de l'application, le pipeline HDR peut être paramétré avec le nombre d'images, et en fonction de la plage dynamique de la scène capturée. Dans la thèse, nous avons présenté deux implémentations différentes, en utilisant respectivement deux ou trois images. Pour chaque mise en œuvre, nous obtenons des résultats performants en raison d'un réglage fin des paramètres des algorithmes de Debevec et un de tone mapping global par Duan. La comparaison avec d'autres architectures issues de l'état de l'art a été faite, et met en évidence une bonne qualité visuelle, à proximité de l'algorithme de Drago, connu comme l'un des meilleurs TMOs.

En outre, pour atteindre la performance temporelle élevée, la plate-forme HDR-ARtiSt intègre une unité de mémoire dédiée à la gestion des flux vidéos. Cette unité de mémoire a été spécialement conçue pour gérer en parallèle plusieurs flux vidéos, afin d'alimenter le processus de création HDR. Il contribue de manière significative à la performance globale du système. En effet, le pipeline HDR est synchronisé sur le framerate du capteur d'image. Il construit une nouvelle image HDR pour chaque nouvelle capture, et délivre un flux en contenu et en direct directement affichable avec une bande passante de  $629\text{Mbits/s}$ . Une telle unité de mémoire limite aussi considérablement le temps de visibi-

lité du flou engendré par le mouvement pendant la capture, et donc les artefacts de type "ghost".

## 6.2/ PERSPECTIVES

Tous ces résultats ouvrent des pistes prometteuses pour l'exploration future des problèmes matériels et logiciels. Les nouvelles versions de la plate-forme sont en prévisions et utiliseront les nouvelles familles de FPGA de Xilinx Virtex-7 et Zynq. Cela nous donnera l'occasion de mettre en œuvre d'autres algorithmes plus récents pour la création des cartes de radiance et le tone mapping local/global. Nous prévoyons également d'évaluer automatiquement la gamme dynamique réelle de la scène, puis d'adapter dynamiquement le nombre de captures nécessaires pour construire la meilleure image HDR. D'autres travaux peuvent découler directement de l'architecture mise en place. Nous verrons tout particulièrement plusieurs pistes de recherche encore inexplorées et à développer, en s'appuyant du travail de cette thèse :

- Concevoir un prototype de A à Z sous Altium (routage FPGA plus aisé) pour s'affranchir des cartes d'évaluation Xilinx
- Développer un algorithme de détection et de suppression de ghosts pour 2, 3 ou 4 images
- Implémenter un algorithme local de tone mapping au sein du pipeline HDR complet
- Aborder les problèmes de "color constancy" pour une implémentation en temps réel
- Développer une caméra rapide HDR à plus de 100 images par seconde.
- Implanter un capteur 2 MégaPixels de chez e2V, sur une carte capteur, compatible pin à pin

La vidéo HDR peut être sujette à des artefacts "ghost", qui peuvent apparaître lorsque le mouvement dans la scène est trop important. Donc, nous aurons l'intention d'étudier et d'implémenter ces techniques de détection de ghost sur FPGA, afin de fournir en temps réel un contenu HDR exploitable dans la plupart des conditions de capture, sans artefacts de ghost.

## 6.3/ EXPÉRIENCE PERSONNELLE

Je ferais une brève conclusion personnelle sur la technique HDR abordée. Bien que la thèse se soit bien déroulée, débouchant sur des choses intéressantes telle qu'une caméra fonctionnelle directement exploitable, je pense que la méthode d'exposition multiple n'est qu'une solution à court terme pour ce qui est de la résolution du problème de dynamique des capteurs actuels. L'enjeu soulevé du domaine HDR, et les intérêts croissants qui vont en faveur de son développement, pousse à croire qu'une nouvelle évolution

technologique aidera prochainement le monde de la recherche à définir de nouvelles approches de conception des systèmes à grande gamme dynamique, notamment dans la définition innovante de technologies implantées à même le capteur.

D'autre part, j'ai eu grande satisfaction à effectuer cette thèse. Du fait de l'aspect technique omniprésent du début à la fin de celle-ci, elle a été pour moi l'occasion de m'initier et de me former aux principaux outils du développeur en électronique et FPGA, de la conception de cartes, au design de systèmes hardware sur FPGAs, etc. J'en garderai très franchement une bonne expérience, la ferai certainement partager dans mes travaux futurs. Malheureusement, j'ai dû faire face à de nombreuses difficultés qui étaient principalement d'ordre technique. J'ai passé beaucoup de temps à réaliser des travaux d'ingénierie, ce qui s'est fait au détriment du temps passé à traiter les aspects de recherche pure, chose que je regrette dans un sens. En d'autres termes, je suis entièrement satisfait des travaux de conception et de design des systèmes que j'ai effectués, mais je suis quelque peu frustré en ayant un sentiment d'inachevé concernant les aspects "théoriques" et conceptuels que ma thèse a pu soulever à certains moments.

De plus, du fait de l'évolution des techniques de design hardware qui sont arrivées en l'espace de mes trois ans de thèse, il m'a paru judicieux d'orienter ma formation technique future vers un design hardware à base de processeurs embarqués (plate-forme zynq par exemple). En effet, je pense que la maîtrise de tels outils pourra permettre de réduire considérablement les temps de développement, de travailler à plus haut niveau dans la hiérarchie, et ainsi envisager d'implémenter des algorithmes beaucoup plus complexes. Certes les débits de données au sein des architectures à bus et à processeur embarqué n'étaient pas suffisants pour le traitement vidéo en temps réel, mais au jour d'aujourd'hui, cette limite semble avoir été dépassée, ce qui rend l'expérience très tentante.

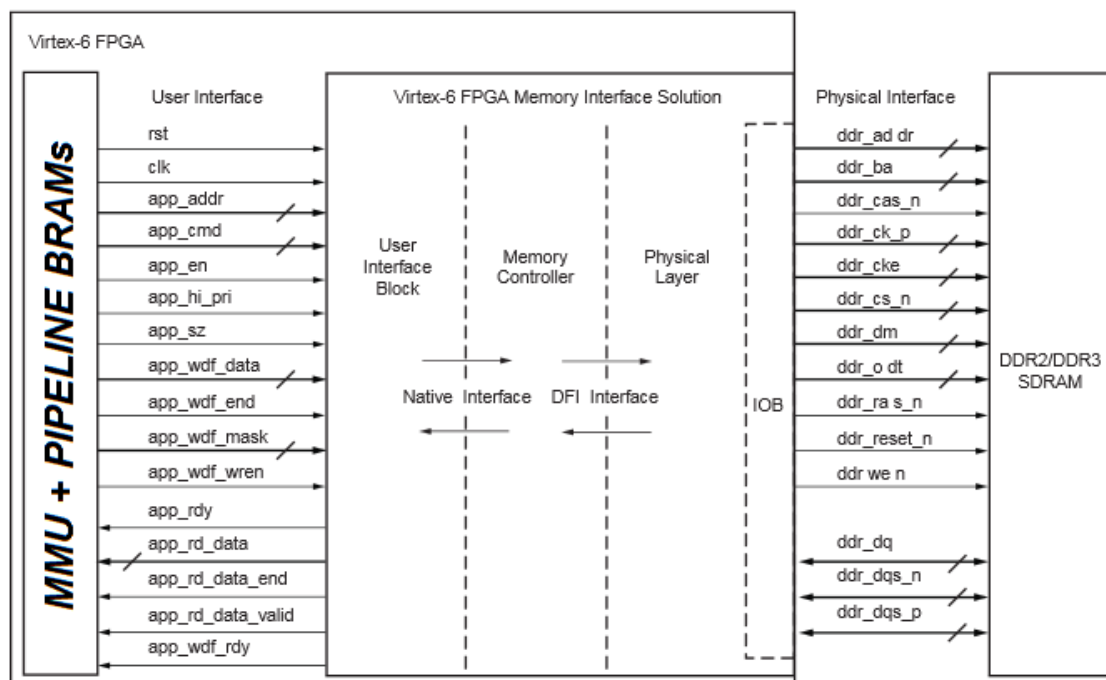


# A

## DESCRIPTION DU MEMORY MANAGEMENT UNIT (MMU) POUR P=3

Nous présenterons ici, par des schémas simplifiés, l'architecture de l'interface mémoire que nous avons développée. Elle est illustrée, à titre d'exemple, à  $P = 3$  images. Celle-ci peut s'étendre à une architecture potentiellement plus vaste en termes de dynamique, utilisant un nombre d'images plus élevé.

La Figure A.1 représente le schéma global d'interconnexion avec la MIG ("Memory Interface Generator") de Xilinx. L'interface utilisateur et les signaux qui lui sont attitrés, sont entièrement générés par nos soins.



**FIGURE A.1** – Schéma global de l'interface mémoire du système HDR-ARtiSt. Elle est composée de la MIG de Xilinx, de notre MMU et du pipeline de BRAMs.

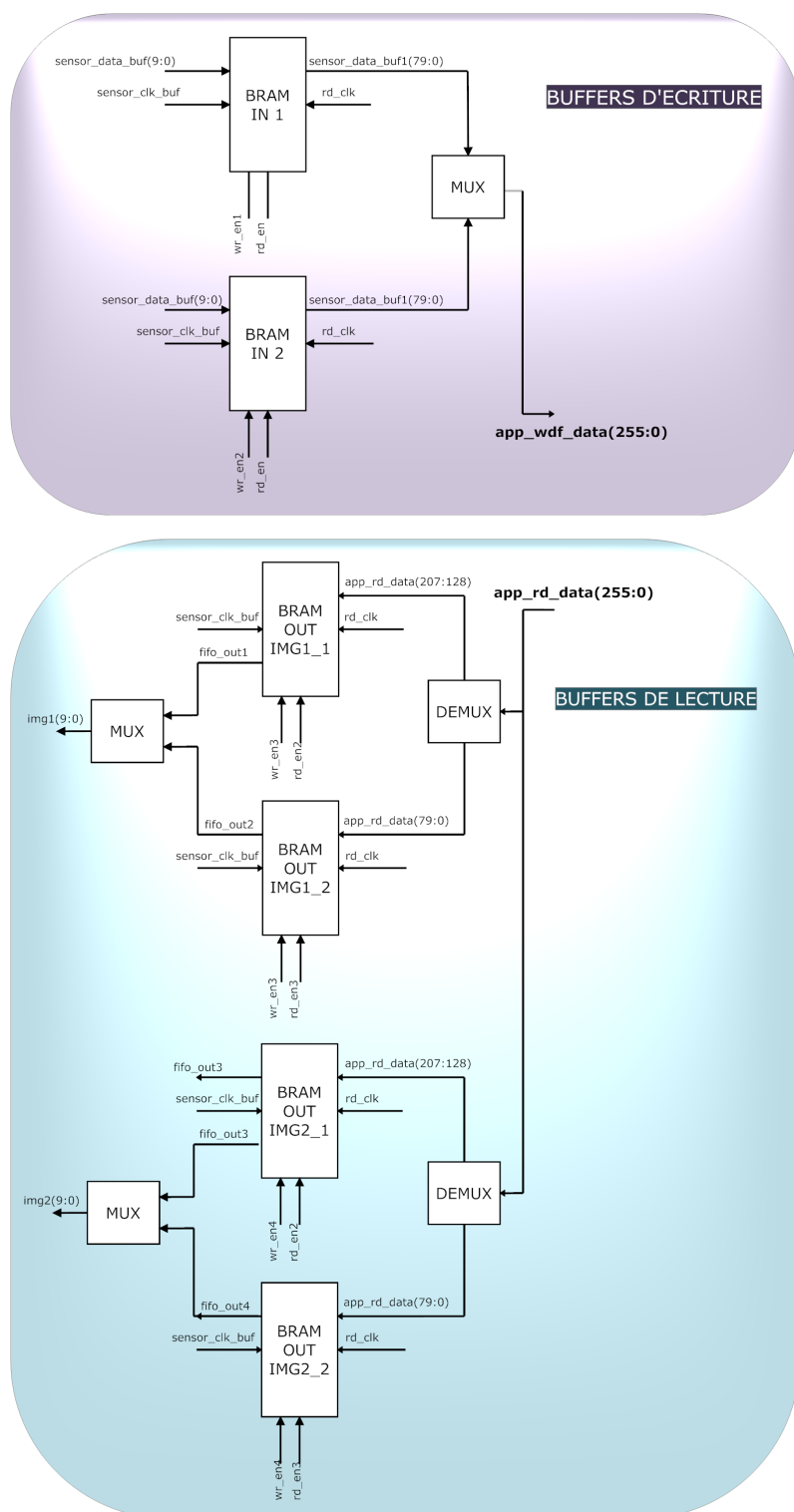
Les buffers d'écriture et de lecture en entrée et en sortie de la MMU sont visibles sur

la Figure A.2. Le nombre de BRAMs nécessaires au fonctionnement du système et à la génération en continue de flux parallélisés et synchronisés est représenté par la fonction suivante :

$$Nb_{BRAMs} = P \times 2 \quad (A.1)$$

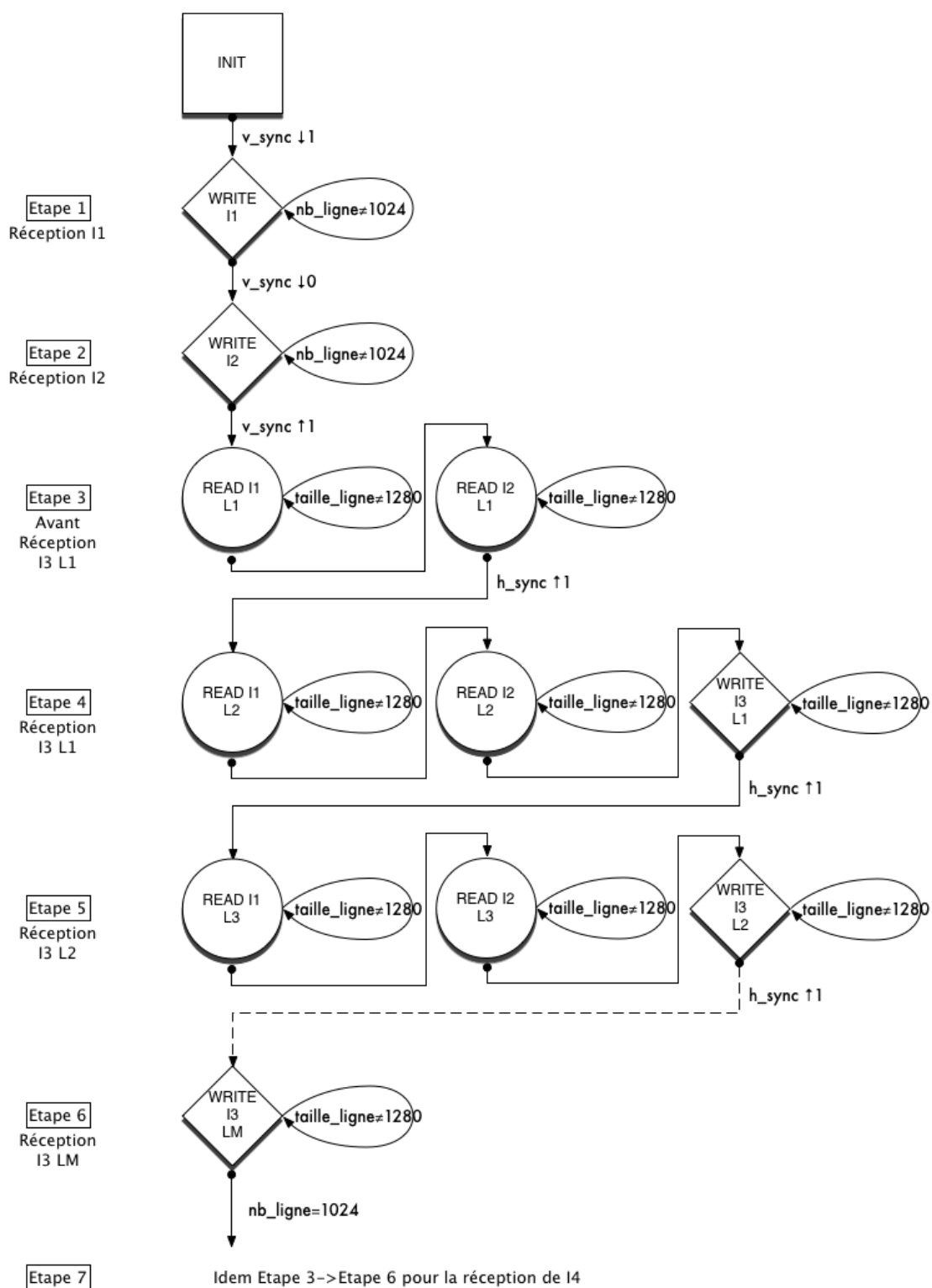
Où  $P$  est le nombre d'images utilisées pour la reconstruction de la carte de radiance. Il correspond en fait au nombre de flux vidéos qui seront parallélisés à l'intérieur du FPGA. Les Blocks RAM ont des domaines d'horloges indépendants en entrées et en sorties. Les deux domaines d'horloges sont ceux du capteur (114MHz environ) et de la mémoire (200MHz). De même, les BRAMs ont des profondeurs de bus d'écriture et de lecture différentes.

- En écriture : Afin d'alimenter et d'optimiser l'utilisation du bus d'écriture de la mémoire en entrée (sur 256bits), on écrit  $2 \times 8$  pixels d'une profondeur de 10bits à chaque coups d'horloge dans le domaine de la mémoire. Pour cela nous sommes obligés de démultiplexer le flux de pixels entrant en deux FIFOs au lieu d'une, pour affecter une taille finale d'entrée du bus de la mémoire à 160bits, correspondant à l'écriture de 16 pixels en un seul coup d'horloge.
- En lecture : Nous désirons avoir une sortie de système à  $P - 1$  flux parallèles synchronisés sur front d'horloge ascendant du capteur, le  $P^{ième}$  flux étant issu du capteur.



**FIGURE A.2** – Schéma du pipeline de BRAMs de la MMU pour  $P = 3$ . Le nombre de BRAMs nécessaires au système général à  $P$  images est de  $2P$ .



FIGURE A.3 – Schéma simplifié de la machine à états de la MMU pour  $P = 3$ .

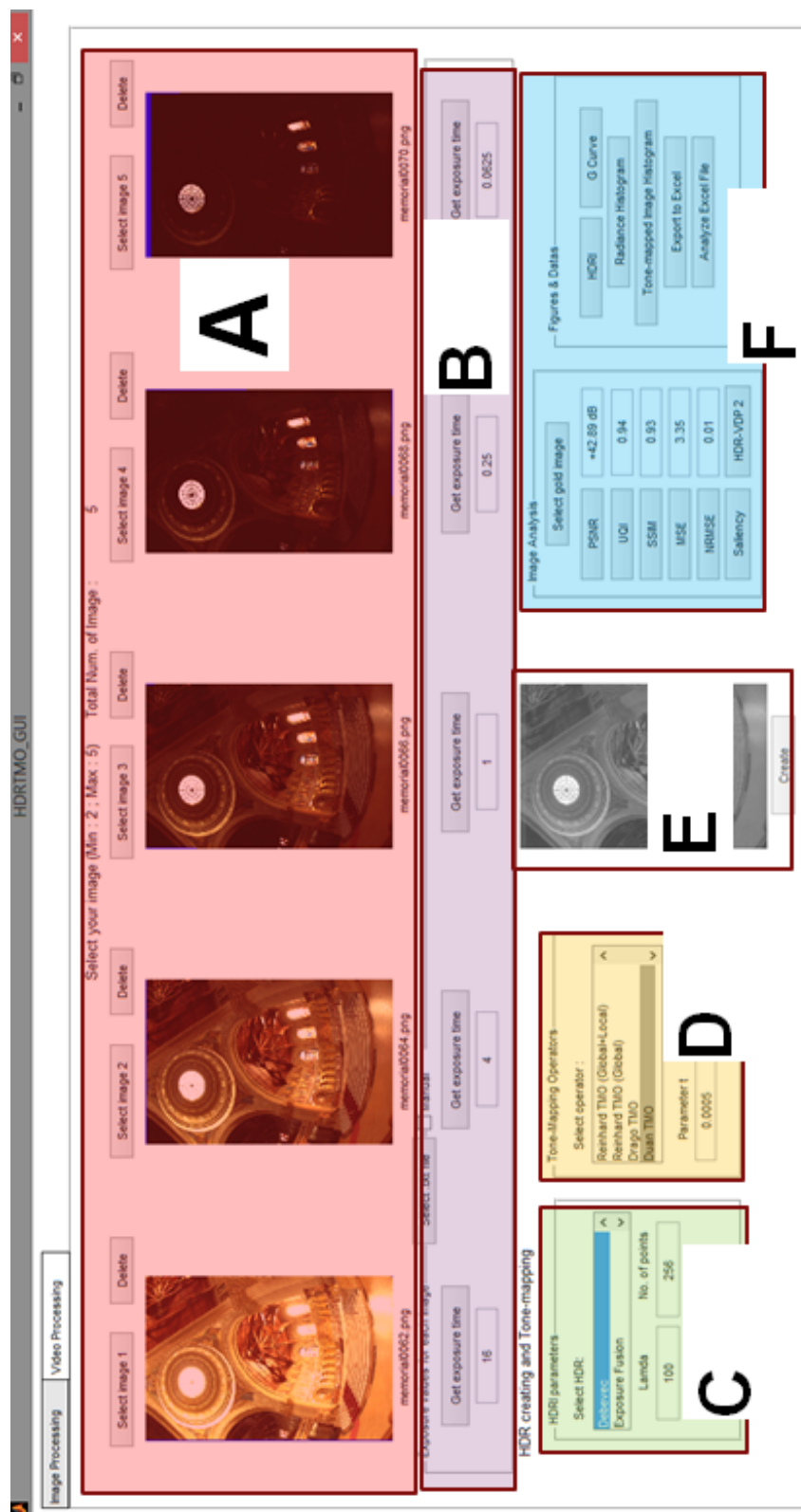
# B

## HDR-LAB

Une interface utilisateur graphique a été développée dans MATLAB. Un GUI permet aux utilisateurs d'effectuer des tâches de manière interactive grâce à des contrôles tels que des boutons et des curseurs. L'outil graphique permet d'effectuer les tâches telles que la création ou la personnalisation d'images HDRs, l'ajustement des courbes de réponse, la visualisation d'images tone mappées, l'analyse et la caractérisation des images, etc. La Figure B.1 montre la mise en page principale de l'application *HDR-Lab*. Il y a quelques éléments majeurs qui sont mis en œuvre dans l'interface graphique. Chacun de ces éléments peut être décomposé. Les éléments concernés sont énumérés ci-dessous, et chacun de ces éléments correspondent à la zone en surbrillance dans la Figure B.1 :

- A. Sélection des images LDRs
- B. Détection des temps d'exposition
- C. Méthode de génération HDR
- D. Tone mapping
- E. Résultat après tone mapping
- F. Analyse des images
- F. Stockage des résultats
- F. Visualisation des résultats, statistiques et métriques qualité

Il a été décidé que le maximum d'images LDRs en entrée serait de cinq, car un trop grand nombre d'images entraînerait un temps de traitement trop élevé.

FIGURE B.1 – Dissection des parties du logiciel *HDR-Lab* sous Matlab.

## ESTIMATION DE LA COURBE DE RÉPONSE

La thèse a débuté par l'étude du capteur d'images, conduisant inévitablement vers un problème bien connu ; la lumière incidente n'est pas retranscrite de façon linéaire en sortie de capteur. En effet la courbe de réponse liant la valeur de pixel  $Z_{ij}$  à la radiance de la scène  $E_{ij}$  est une caractéristique essentielle au domaine HDR, puisque le but finale est d'avoir une vraie représentation des valeurs physiques de luminosités d'une scène capturée. Pour cela, Debevec et al. [10] a trouvé un moyen de retrouver cette courbe à l'aide d'exposition multiple d'une même scène. Les images doivent être bien évidemment parfaitement alignées pour faire les correspondances entre les pixels. Il est aussi nécessaire de connaître les temps d'exposition individuels des images servant à l'évaluation de la courbe. Le code qui suit est issu de la revue de Debevec.

```
1 % On a une série de pixels observés pour plusieurs
2 % images avec des temps d'exposition différents.
3 % La fonction retourne la courbe de réponse du système
4 % et le logarithme des irradiances des pixels correspondants
5 % Zmin = 0 est la valeur minimale des pixels d'entrée
6 % Zmax = 1023 est la valeur maximale des pixels d'entrée
7 %
8 % Arguments:
9 %
10 % Z(i,j) est la valeur du pixel à la position i, dans l'image j
11 % B(j) est le logarithm de l'exposition correspondant à l'image j
12 % l est lambda, la constante de lissage
13 % w(z) is the weighting function value for pixel value z
14 %
15 % La fonction retourne:
16 %
17 % g(z) la valeur d'exposition associée au pixel z
18 % lE(i) l'irradiance du pixel z à la position i
19 %
```

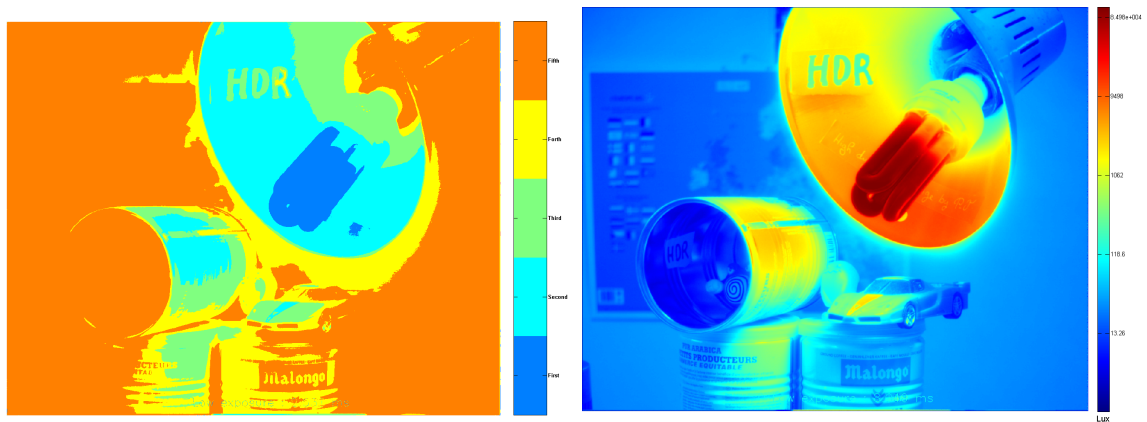
```

20 function [g,lE]=gsolve(Z,B,l,w)
21 n = 256;
22 A = zeros(size(Z,1)*size(Z,2)+n+1,n+size(Z,1));
23 b = zeros(size(A,1),1);
24 k = 1;
25 for i=1:size(Z,1)
26     for j=1:size(Z,2)
27         wij = w(Z(i,j)+1);
28         A(k,Z(i,j)+1) = wij; A(k,n+i) = ?wij; b(k,1) = wij * B(i,j);
29         k=k+1;
30     end
31 end
32 % modifie la courbe pour faire
33 % correspondre sa valeur intermédiaire à 0
34 A(k,129) = 1;
35 k=k+1;
36 % lissage de la courbe
37 for i=1:n?2
38     A(k,i)=l*w(i+1); A(k,i+1)=?2*1*w(i+1); A(k,i+2)=l*w(i+1);
39     k=k+1;
40 end
41 % "Single value decomposition"
42 x = A\b;
43 g = x(1:n);
44 lE = x(n+1:size(x,1));

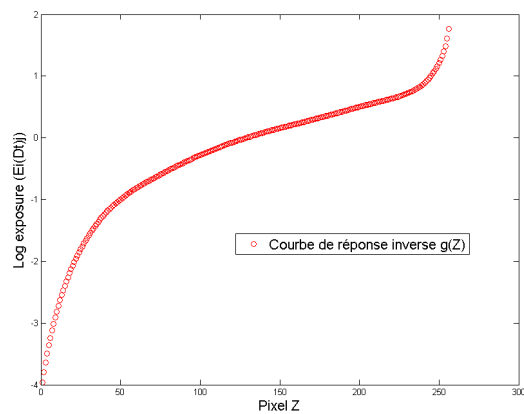
```



(a)



(b)



(c)

**FIGURE C.1** – Images LDRs issues de notre camera *HDR-ARtiSt*, utilisées afin d'estimer la courbe de réponse du système à partir de logiciel *HDR-Lab*. Les temps d'exposition sont :  $16\mu s$ ,  $110\mu s$ ,  $629\mu s$ ,  $2.53ms$  et  $9.00ms$ . Les images de contribution et fausse couleur associées sont représentées (f) et (g).



D

"HDR-VIEWFINDER" : LES SORTIES  
DU SYSTÈME HDR-ARTIST





FIGURE D.1 – Les différentes sorties parallélisées du système HDR-ARTiSt.

E

ÉTALON OR



**FIGURE E.1** – L'étalon or pour les comparaisons d'algorithmes de tone mapping sous HDR-Lab. C'est une version d'image HDR créée à partir de 16 images LDRs, et tone mappée avec l'algorithme de Drago et al. [13].

## DÉTAILS DES CARTES CAPTEUR

Nous présentons ici les schémas de la carte fille et sous-fille F.1F.2, faits avec le logiciel Orcad Capture. A titre indicatif, nous définirons la liste des composants.

### F.1/ LISTE DES COMPOSANTS

#### F.1.1/ CARTE SOUS-FILLE

##### F.1.1.1/ L'OPTIQUE

- Lentille : M12 Sunex lens DSL901C-NIR
- Support cmt103 ou
- cmt1205C(adaptateur)

##### F.1.1.2/ L'ÉLECTRONIQUE

###### **Pour l'entourage du capteur :**

- 7 condensateurs céramique X7R non polarisés de 100nF
- 3 condensateurs polarisés de 10F
- 1 résistance Rext de 16.9kΩ
- 4 connecteurs HARWIN - M40-6201546 - FICHE FEMELLE CMS 1.00MM 15 VOIES

#### F.1.2/ CARTE FILLE

##### F.1.2.1/ L'ÉLECTRONIQUE

- 3 SN74AVC8T245 pour les bus de données
- 3 TPS54310 pour générer du 3.3V, 1.8V Analogique et 1.8V à partir du 5V  
OU 2 composants TPS54314 (5V → 1.8V et 1 TPS54316 (5V → 3.3V)
- 4 connecteurs HARWIN - M40-6001546 - FICHE MALE CMS 1.25MM 15 VOIES
- 1 oscillateur à Quartz de 27MHz + 1 condensateur 0.01F

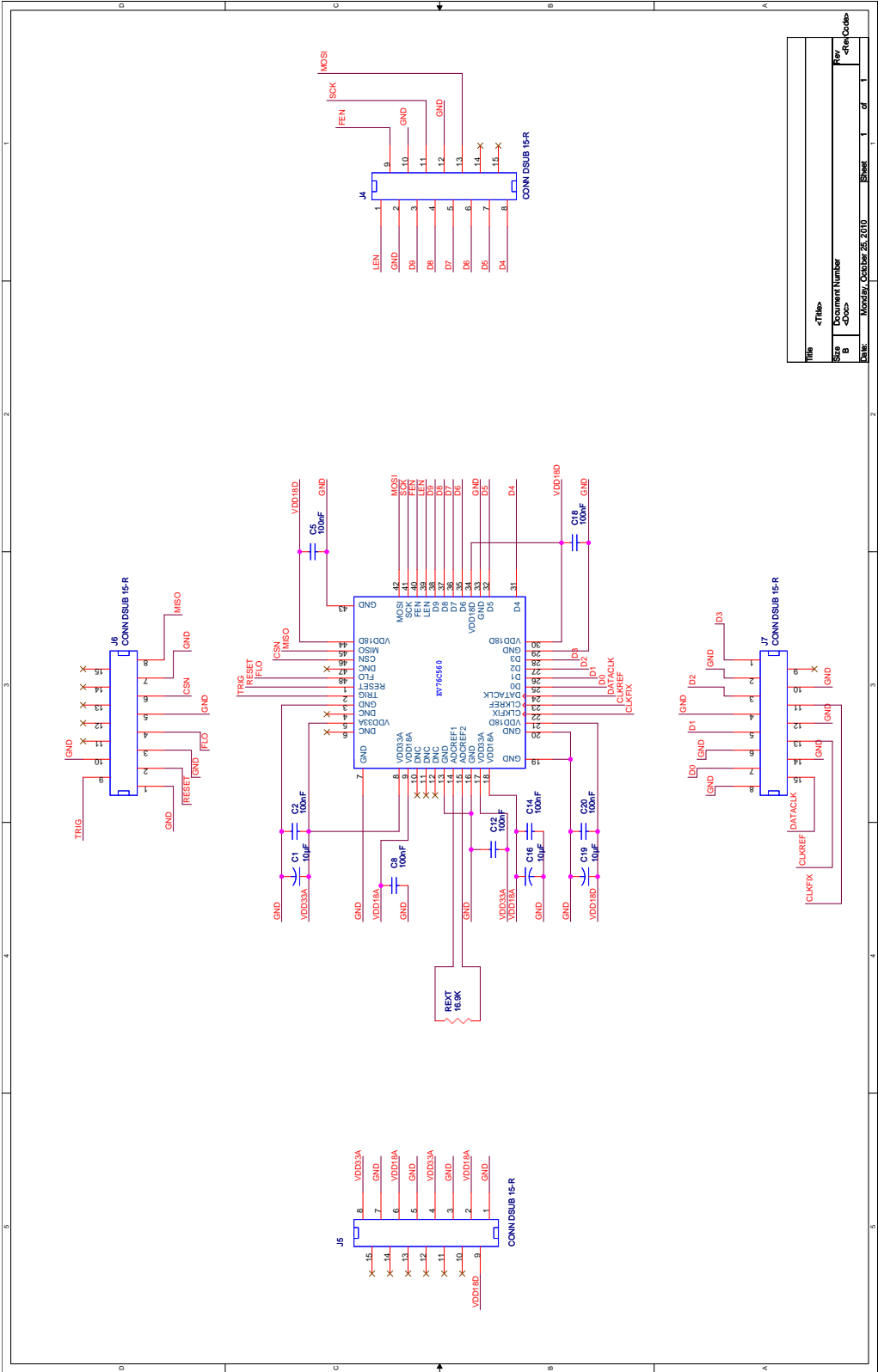
- 1 connecteur HARTING - 0973 296 6801 - DIN41612 FEMELLE R96

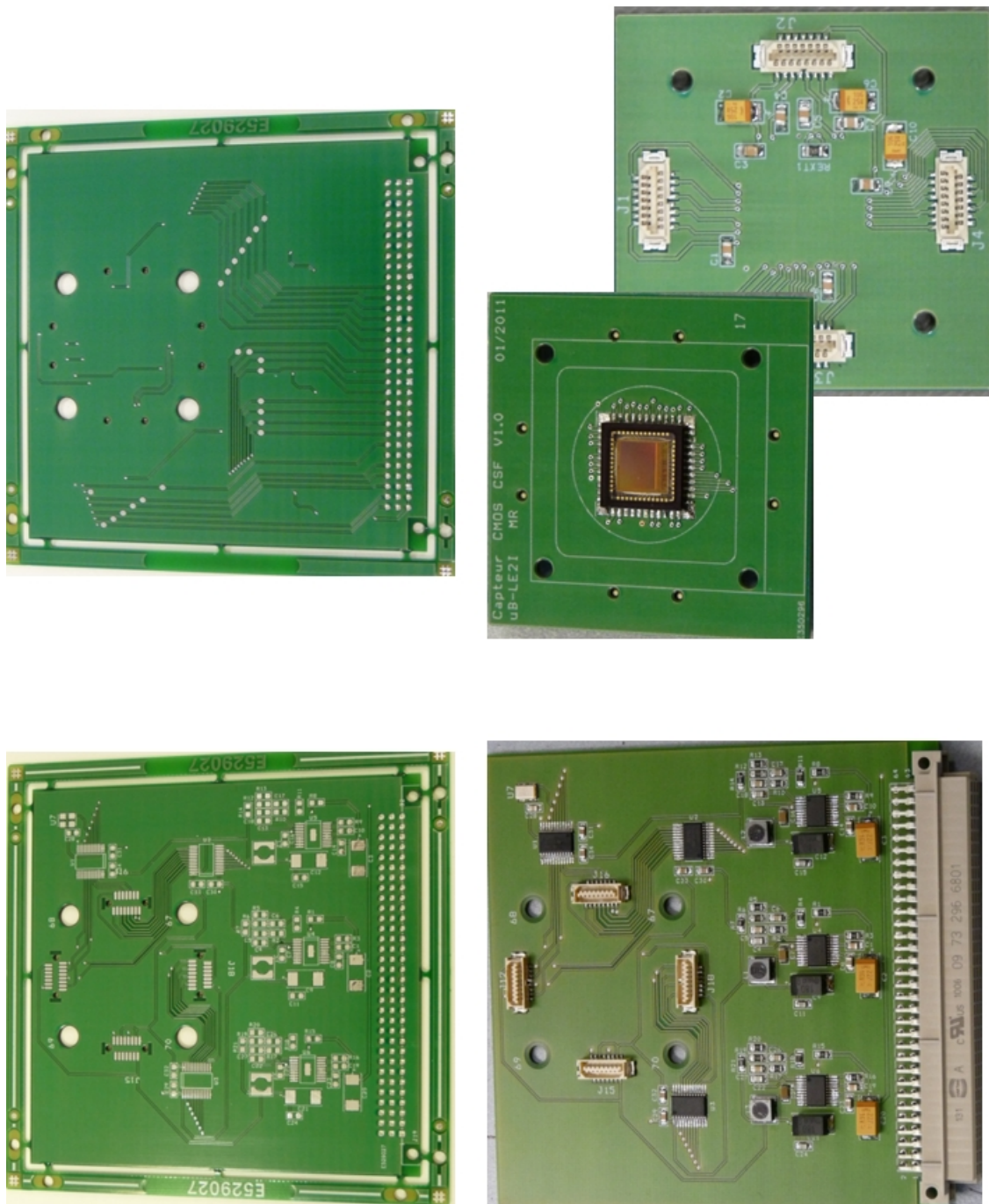
**Pour UN TPS54310 :**

- Condensateur C2 polarisé 220F POSCAP max 3A
- C3 0.1F
- C4 100pF
- C5 3900pF
- C6 2700pF
- C7 0.047F
- C8 10F
- C9 polarisé 180F 4V ESR < 0.015Ω
- C11 1000pF
- L1 1.2H Coilcraft DO1813P-122HC 0.017 Ω
- Résistance R1 10kΩ
- R2 3.74kΩ
- R3 71.5kΩ
- R4 3.74kΩ ou 9.76kΩ suivant la tension Vout désirée
- R5 10kΩ
- R6 732Ω
- R7 49.9Ω



**FIGURE F.1** – Schéma de la carte fille destinée à être montée sur des connecteurs XGI.





**FIGURE F.3** – Vues recto verso des cartes capteur fille et sous-fille.





# BIBLIOGRAPHIE

- [1] A. Spivak, A. Belenky, A. Fish, and O. Yadid-Pecht. Wide-dynamic-range cmos image sensors—comparative performance analysis. *Electron Devices, IEEE Transactions on*, vol. 56, no. 11(11) :2446–2461, nov. 2009.
- [2] S. Kavadias, B. Dierickx, D. Scheffer, A. Alaerts, D. Uwaerts, and J. Bogaerts. A logarithmic response cmos image sensor with on-chip calibration. *Solid-State Circuits, IEEE Journal of*, 35(8) :1146–1152, 2000.
- [3] S.K. Nayar and T. Mitsunaga. High dynamic range imaging : spatially varying pixel exposures. In *Computer Vision and Pattern Recognition. Proceedings. IEEE Conference on*, volume 1, pages 472–479 vol.1, 2000.
- [4] P. Acosta-Serafini, M. Ichiro, and C. Sodini. A 1/3" VGA linear wide dynamic range CMOS image sensor implementing a predictive multiple sampling algorithm with overlapping integration intervals. *IEEE Journal of Solid-State Circuits*, 39(9) :1487–1496, Sept 2004.
- [5] M. Sakakibara, S. Kawahito, D. Handoko, N. Nakamura, M. Higashi, K. Mabuchi, and H. Sumi. A high-sensitivity CMOS image sensor with gain-adaptative column amplifiers. *IEEE Journal of Solid-State Circuits*, 40(5) :1147–1156, May 2005.
- [6] G.L. Cembrano, A. Rodriguez-Vazquez, R.C. Galan, F. Jimenez-Garrido, S. Espejo, and R. Dominguez-Castro. A 1000 fps at  $128 \times 128$  vision processor with 8-bit digitized I/O. *IEEE Journal of Solid-State Circuits*, 39(7) :1044–1055, Jul 2004.
- [7] L. Lindgren, J. Melander, R. Johansson, and B. MÅČÅüller. A multiresolution 100-GOPS 4-Gpixels/s programmable smart vision sensor for multisense imaging. *IEEE Journal of Solid-State Circuits*, 40(6) :1350–1359, Jun 2005.
- [8] Saverio Morfu, Patrick Marquié, Brice Nofiélé, and Dominique Gin hac. Nonlinear systems for image processing. In Peter W. Hawkes, editor, *Advances in Imaging and Electron Physics*, volume 152, pages 79 – 151. Elsevier, 2008.
- [9] J. Dubois, D. Gin hac, M. Paindavoine, and B. Heyrman. A 10 000 fps CMOS sensor with massively parallel image processing. *IEEE Journal of Solid-State Circuits*, 43(3) :706–717, Mar 2008.
- [10] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH*, pages 369–378, 1997.
- [11] Mark A. Robertson, Sean Borman, and Robert L. Stevenson. Estimation-theoretic approach to dynamic range enhancement using multiple exposures. *Journal of Electronic Imaging*, 12(2) :219–228, 2003.

- [12] T. Mitsunaga and S.K. Nayar. Radiometric Self Calibration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 374–380, Jun 1999.
- [13] T. Annen F. Drago, K. Myszkowski and N. Chiba. Adaptive logarithmic mapping for displaying high contrast scenes. *EUROGRAPHICS 2003 / P. Brunet and D. Fellner*, Volume 22 (2003), Number 3(3) :xx – xx, 2003.
- [14] E Reinhard, M Stark, P Shirley, and J Ferwerda. Photographic tone reproduction for digital images. *ACM Transactions on Graphics*, 21(3) :267–276, 2002.
- [15] Fr  do Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. Technical report, Laboratory for Computer Science, Massachusetts Institute of Technology, 2002.
- [16] T. K. Castro, A. Chapiro, M. Cicconet, and L. Velho. Towards Mobile HDR Video. In *Eurographics*, pages 75–76, 2011.
- [17] Michael Werman Raanan Fattal, Dani Lischinski. Gradient domain high dynamic range compression. Technical report, School of Computer Science and Engineering The Hebrew University of Jerusalem, 2002.
- [18] J. Dubois, D. Ginhac, M. Paindavoine, and B. Heyrman. A 10 000 fps cmos sensor with massively parallel image processing. *Solid-State Circuits, IEEE Journal of*, 43(3) :706–717, March.
- [19] Fan Yang and M. Paindavoine. Implementation of an rbf neural network on embedded systems : real-time face tracking and identity verification. *Neural Networks, IEEE Transactions on*, 14(5) :1162–1175, Sept.
- [20] N. Farrugia, F. Mamalet, S. Roux, Fan Yang, and M. Paindavoine. Fast and robust face detection on a parallel optimized architecture implemented on fpga. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(4) :597–602, April.
- [21] Photomatix. [http ://www.hdrsoft.com/fr/](http://www.hdrsoft.com/fr/).
- [22] Enfuse. [http ://enblend.sourceforge.net/](http://enblend.sourceforge.net/).
- [23] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec. *High Dynamic Range Imaging : Acquisition, Display, and Image-Based Lighting (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.
- [24] M.S. Sayed and J.G.R. Delva. An efficient intensity correction algorithm for high definition video surveillance applications. *Circuits and Systems for Video Technology, IEEE Transactions on*, 21(11) :1622 –1630, nov. 2011.
- [25] Alberto Boschetti, Nicola Adami, Riccardo Leonardi, and Masahiro Okuda. An optimal Video-Surveillance approach for HDR videos tone mapping. In *19th European Signal Processing Conference EUSIPCO 2011*, Barcelona, August 2011.

- [26] S. Mangiat and J. Gibson. Inexpensive high dynamic range video for large scale security and surveillance. In *Military Communications Conference, 2011 - MILCOM 2011*, pages 1772 –1777, nov. 2011.
- [27] Suk-Ho Lee, Hyenkyun Woo, and Moon Gi kang. Global illumination invariant object detection with level set based bimodal segmentation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(4) :616 –620, april 2010.
- [28] M. Schanz, C. Nitta, A. Bussmann, B.J. Hosticka, and R.K. Wertheimer. A high-dynamic-range cmos image sensor for automotive applications. *Solid-State Circuits, IEEE Journal of*, 35(7) :932 –938, july 2000.
- [29] Xianbin Cao, Changxia Wu, Jinhe Lan, Pingkun Yan, and Xuelong Li. Vehicle detection and motion analysis in low-altitude airborne video under urban environment. *Circuits and Systems for Video Technology, IEEE Transactions on*, 21(10) :1522 –1533, oct. 2011.
- [30] Stefan Hrabar, Peter Corke, and Michael Bosse. High dynamic range stereo vision for outdoor mobile robotics. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 430 –435, may 2009.
- [31] P.-F. Ruedi, P. Heim, S. Gyger, F. Kaess, C. Arm, R. Caseiro, J.-L. Nagel, and S. Todeschini. An soc combining a 132db qvga pixel array and a 32b dsp/mcu processor for vision applications. In *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pages 46 –47,47a, feb. 2009.
- [32] Meagan Leflar, Omar Hesham, and Chris Joslin. Use of high dynamic range images for improved medical simulations. In Nadia Magnenat-Thalmann, editor, *Modeling the Physiological Human*, volume 5903 of *Lecture Notes in Computer Science*, pages 199–208. Springer Berlin Heidelberg, 2009.
- [33] H.-G. Graf, C. Harendt, T. Engelhardt, C. Scherjon, K. Warkentin, H. Richter, and J.N. Burghartz. High dynamic range cmos imager technologies for biomedical applications. *Solid-State Circuits, IEEE Journal of*, 44(1) :281 –289, jan. 2009.
- [34] Mehrdad Abolbashari, Gelareh Babaie, Filipe Magalhães, Miguel V. Correia, Francisco M. Araújo, Awad S. Gerges, and Faramarz Farahi. Biological imaging with high dynamic range using compressive imaging technique. In *Proc. SPIE 8225, Imaging, Manipulation, and Analysis of Biomolecules, Cells, and Tissues X*, volume 8225, pages 82251X–82251X–7, 2012.
- [35] M. Kfoury, O. Marinov, P. Quevedo, N. Faramarzpour, S. Shirani, L.W.-C. Liu, Qiyin Fang, and M.J. Deen. Toward a miniaturized wireless fluorescence-based diagnostic imaging system. *Selected Topics in Quantum Electronics, IEEE Journal of*, 14(1) :226 –234, jan.-feb. 2008.
- [36] Frederic Leblond, Scott C. Davis, Pablo A. Vald  s, and Brian W. Pogue. Pre-clinical whole-body fluorescence imaging : Review of instruments, methods and

- applications. *Journal of Photochemistry and Photobiology B : Biology*, 98(1) :77 – 94, 2010.
- [37] Nicholas B Mackinnon and Ulrich Stange. Apparatus and methods relating to expanded dynamic range imaging endoscope systems, October 11 2006. EP Patent 1,709,476.
- [38] T.H. Khan and K.A. Wahid. Low power and low complexity compressor for video capsule endoscopy. *Circuits and Systems for Video Technology, IEEE Transactions on*, 21(10) :1534 –1546, oct. 2011.
- [39] Eric R. Fossum. Active pixel sensors : are ccds dinosaurs? *Proc. SPIE 1900, Charge-Coupled Devices and Solid State Optical Sensors III*, 2 :2–14, 1993.
- [40] Mann, Picard, S. Mann, and R. W. Picard. On being ‘undigital’ with digital cameras : Extending dynamic range by combining differently exposed pictures. In *Proceedings of IS&T*, pages 442–448, 1995.
- [41] Thor Olson. High dynamic range astronomical imaging. In *Fifteenth Color Imaging Conference : Color Science and Engineering Systems, Technologies, and Applications*, 2007.
- [42] W. Stuerzlinger G. Yourganov. Acquiring high dynamic range video at video rates. Technical report, Acquiring High Dynamic Range Video at Video Rates, 2001.
- [43] Asli Gençtav and Ahmet Oğuz Akyüz. Evaluation of radiometric camera response recovery methods. In *SIGGRAPH Asia 2011 Posters*, SA ’11, pages 15 :1–15 :1, New York, NY, USA, 2011. ACM.
- [44] Daniel J. Jobson, Zia ur Rahman, and Glenn A. Woodell. Retinex image processing : Improved fidelity to direct visual observation, 1996.
- [45] Sumanta N. Pattanaik, Jack Tumblin, Hector Yee, and Donald P. Greenberg. Time-dependent visual adaptation for fast realistic image display. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH ’00, pages 47–54, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [46] Mark D. Fairchild and Garrett M. Johnson. Meet icam : A next-generation color appearance model. In *IS&T/SID 10 th Color Imaging Conference*, pages 33–38, 2002.
- [47] K Chiu, M Herf, P Shirley, S Swamy, C Wang, and K Zimmerman. Spatially nonuniform scaling functions for high contrast images. In *In Proceedings of Graphics Interface 93*, pages 245–253, 1993.
- [48] Yangli Hector Yee and Sumanta Pattanaik. Segmentation and adaptive assimilation for detail-preserving display of high-dynamic range images. *The Visual Computer*, 19(7-8) :457–466, 2003.
- [49] Firas Hassan and Joan Carletta. A real-time fpga-based architecture for a reinhard-like tone mapping operator. In *gh2007*, 2007.

- [50] ChrisDance Jiang Duan, MarcoBressan and GuopingQiu. Tone-mapping high dynamic range images by novel histogram adjustment. *Pattern Recognition*, 43 :1847–1862, 2010.
- [51] Alessandro Artusi, Jiří Bittner, Michael Wimmer, and Alexander Wilkie. Delivering interactivity to complex tone mapping operators. In *Proceedings of the 14th Eurographics workshop on Rendering*, EGRW '03, pages 38–44, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [52] Rafal Mantiuk, Grzegorz Krawczyk, Radosław Mantiuk, and Hans-Peter Seidel. High dynamic range imaging pipeline : Perception-motivated representation of visual content. Technical report, MPI Informatik, Szczecin University of Technology, 2007.
- [53] Michael Ashikhmin. A tone mapping algorithm for high contrast images. In *Proceedings of the 13th Eurographics workshop on Rendering*, EGRW '02, pages 145–156, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.
- [54] Erik Reinhard and Kate Devlin. Dynamic range reduction inspired by photoreceptor physiology. *IEEE Transactions on Visualization and Computer Graphics*, 11 :13 – 24, 2005.
- [55] Karol Myszkowski Rafal Mantiuk and Hans-Peter Seidel. A perceptual framework for contrast processing of high dynamic range images. Technical report, MPI Informatik, 2006.
- [56] Ahmet Oğuz Akyüz and Erik Reinhard. Perceptual evaluation of tone-reproduction operators using the cornsweet–craik–o'brien illusion. *ACM Trans. Appl. Percept.*, 4 :1 :1–1 :29, February 2008.
- [57] Frederic Drago, William Martens, Karol Myszkowski, and Hans-Peter Seidel. Perceptual evaluation of tone mapping operators with regard to similarity and preference. Research Report MPI-I-2002-4-002, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, August 2002.
- [58] Martin Cadík. Evaluation of hdr tone mapping methods using essential perceptual attributes. *Computers & Graphics*, 32(6) :716–719, 2008.
- [59] Patrick Ledda, Alan Chalmers, Tom Troscianko, and Helge Seetzen. Evaluation of tone mapping operators using a high dynamic range display. In *ACM SIGGRAPH 2005, LA*. ACM Press, August 2005.
- [60] Garrett M. Johnson and Mark D. Fairchild. Rendering hdr images. In *in IS&T/SID 11th Color Imaging Conference*, pages 36–41, 2003.
- [61] Gregory Ward Larson, Gregory Ward Larson, Holly Rushmeier, Holly Rushmeier, Christine Piatko, and Christine Piatko. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics*, 3 :291–306, 1997.

- [62] Patrick Ledda, Luis Paulo dos Santos, and Alan Chalmers. A local model of eye adaptation for high dynamic range. In *Proceedings of the 3rd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa. AFRIGRAPH2004*. ACM Press, November 2004.
- [63] Akiko Yoshida, Volker Blanz, Karol Myszkowski, and Hans peter Seidel. Perceptual evaluation of tone mapping operators with real-world scenes. In *Human Vision & Electronic Imaging X, SPIE*, pages 192–203. SPIE, 2005.
- [64] Frank Van Reeth Tom Mertens, Jan Kautz. Exposure fusion. In *Exposure Fusion*, 2007.
- [65] A. Ardeshir Goshtasby. Fusion of multi-exposure images. *Image and Vision Computing*, 23 :611–618, 2005.
- [66] Frank Van Reeth Tom Mertens, Jan Kautz. Exposure fusion. Technical report, Hasselt University EDM transationale Universiteit Limburg Belgium, University College London UK, 2007.
- [67] Fabrizio Pece and Jan Kautz. Bitmap movement detection : Hdr for dynamic scenes. *Visual Media Production, Conference for*, 0 :1–8, 2010.
- [68] Greg Ward. Fast, robust image registration for compositing high dynamic range photographs from hand-held exposures. *journal of graphics, gpu, and game tools*, 8(2) :17–30, 2003.
- [69] Katrien Jacobs, Celine Loscos, and Greg Ward. Automatic high-dynamic range image generation for dynamic scenes. *IEEE Computer Graphics and Applications*, 28(2) :84–93, 2008.
- [70] Dro Desire Sidibe, William Puech, and Olivier Strauss. Ghost Detection and Removal in High Dynamic Range Images. In *European Signal Processing Conference*, page 5, Glasgow, Royaume-Uni, August 2009. 5 pages.
- [71] Tae-Hong Min, Rae-Hong Park, and Soonkeun Chang. Histogram based ghost removal in high dynamic range images. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 530–533, 2009.
- [72] Abhilash Srikantha and DÃ’sirÃ’ SidibÃ’. Ghost detection and removal for high dynamic range images : Recent advances. *Signal Processing : Image Communication*, 27(6) :650 – 662, 2012.
- [73] Christian Bloch. *The HDRI Handbook : High Dynamic Range Imaging for Photographers and CG Artists +DVD*. Rocky Nook, 2007.
- [74] S. Decker, R.D. McGrath, K. Brehmer, and C.G. Sodini. A 256 times ;256 cmos imaging array with wide dynamic range pixels and column-parallel digital output. *Solid-State Circuits, IEEE Journal of*, 33(12) :2081–2091, 1998.
- [75] Peng Lin Huai Dong Li. Imaging systems and methods for generating motion-compensated high-dynamic-range images, 2013.

- [76] Xiaochuan Guo, Xin Qi, and J.G. Harris. A time-to-first-spike cmos image sensor. *Sensors Journal, IEEE*, 7(8) :1165–1175, Aug.
- [77] Bogdan Arion Yang Ni, YiMing Zhu. A 768x576 logarithmic image sensor with photodiode in solar cell mode. In *International Image Sensor Workshop*, 2011.
- [78] Abbas El Gamal. High dynamic range image sensors. In *ISSCC*, 2002.
- [79] Asheer Kasar Bachoo. Real-time exposure fusion on a mobile computer. Technical report, Signal Processing Research Group Optronics Sensor Systems Council for Scientific and Industrial Research (CSIR) Pretoria, South Africa, 2009.
- [80] Marius Tico, Natasha Gelfand, and Kari Pulli. Motion-blur-free exposure fusion. In *ICIP'10*, pages 3321–3324, 2010.
- [81] Won-ho Cho Seung-Jun Youm and Ki-Sang Hong. High dynamic range video through fusion of exposed-controlled frames. In *Proceedings of IAPR Conference on Machine Vision Applications*, pages 546–549, 2005.
- [82] Michael D. Tocci, Chris Kiser, Nora Tocci, and Pradeep Sen. A Versatile HDR Video Production System. *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2011)*, 30(4) :9, 2011.
- [83] B. Guthier, S. Kopf, and W. Effelsberg. Optimal shutter speed sequences for real-time hdr video. In *Imaging Systems and Techniques (IST), 2012 IEEE International Conference on*, pages 303 –308, july 2012.
- [84] Benjamin Guthier, Stephan Kopf, and Wolfgang Effelsberg. Algorithms for a real-time hdr video system. *Pattern Recognition Letters*, 34(1) :25 – 33, 2013. Extracting Semantics from Multi-Spectrum Video.
- [85] M.A. Ali and S. Mann. Comparametric image compositing : Computationally efficient high dynamic range imaging. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 913 –916, march 2012.
- [86] Luiz Velho. Histogram-based hdr video. In *ACM SIGGRAPH 2007 posters*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.
- [87] M.A. Robertson, S. Borman, and R.L. Stevenson. Dynamic range improvement through multiple exposures. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 3, pages 159–163 vol.3, 1999.
- [88] Firas Hassan and Joan E. Carletta. An fpga-based architecture for a local tone-mapping operator. *J Real-Time Image Proc*, 2 :293–308, 2007.
- [89] P Debevec, E Reinhard, G Ward, and S Pattanaik. High dynamic range imaging. *ACM SIGGRAPH 2004 Course Notes*, 13 :14, 2004.
- [90] advisor : Dr. Joan E. Carletta Lavanya Vytla. A real-time implementation of gradient domain high dynamic range compression using a local poisson solver. Master's thesis, The Graduate Faculty of The University of Akron, 2010.



- [91] Mike Tocci Chris Kiser, Erik Reinhard and Nora Tocci. Real-time automated tone mapping system for hdr video. In *IEEE International Conference on Image Processing*, 2012.
- [92] Ching-Te Chiu, Tsun-Hsien Wang, Wei-Ming Ke, Chen-Yu Chuang, Jhih-Siao Huang, Wei-Su Wong, Ren-Song Tsay, and Cyuan-Jhe Wu. Real-time tone-mapping processor with integrated photographic and gradient compression using 0.13um technology on an arm soc platform. *Journal of Signal Processing Systems*, pages 1–15, 2010. 10.1007/s11265-010-0491-8.
- [93] Simon Winder Sing Bing Kang, Matthew Uyttendaele and Richard Szeliski. High dynamic range video. Technical report, Interactive Visual Media Group, Microsoft Research, Redmond, WA, 2003.
- [94] Natasha Gelfand, Andrew Adams, Sung Hee Park, and Kari Pulli. Multi-exposure imaging on mobile devices. In *Proceedings of the international conference on Multimedia*, MM '10, pages 823–826, New York, NY, USA, 2010. ACM.
- [95] Jonas Unger and Stefan Gustavson. High-dynamic-range video for photometric measurement of illumination. *Proceedings of the SPIE*, pages 65010E–65010E–10, 2007.
- [96] Raquel Ureña, Pablo Martinez-Cañada, Juan Manuel Gómez-López, Christian A. Morillas, and Francisco J. Pelayo. Real-time tone mapping on gpu and fpga. *EURASIP J. Image and Video Processing*, 2012 :1, 2012.
- [97] Ahmet Ö. Akyüz. High dynamic range imaging pipeline on the gpu. *Journal of Real-Time Image Processing*, pages 1–15, 2012.
- [98] Wei-Ming Ke, Tsun-Hsien Wang, and Ching-Te Chiu. Hardware-efficient virtual high dynamic range image reproduction. In *Proceedings of the 16th IEEE International Conference on Image Processing (ICIP'09)*, pages 2665–2668, Piscataway, NJ, USA, 2009. IEEE Press.
- [99] S. Mann, R.C.H. Lo, K. Ovtcharov, Shixiang Gu, D. Dai, C. Ngan, and Tao Ai. Realtime hdr (high dynamic range) video for eyetap wearable computers, fpga-based seeing aids, and glasseyes (eyetaps). In *Electrical Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*, pages 1 –6, 29 2012-may 2 2012.
- [100] Benjamin Guthier, Stephan Kopf, and Wolfgang Effelsberg. A real-time system for capturing hdr videos. In *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pages 1473–1476, New York, NY, USA, 2012. ACM.
- [101] e2v technologies. Ev76c560 BW and colour CMOS sensor, 2009.
- [102] Maruyama T. Asano S. and Yamaguchi Y. Performance comparison of fpga, gpu and cpu in image processing. In *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on*, pages 126–131, 2009.

- [103] Inc. Zynq-7000 all programmable SoC overview, August 2012.
- [104] Inc. ZedBoard (Zynq evaluation and development) hardware user's guide, September 2012.
- [105] L.E Alston, D.S. Levinstone, and W.T. Plummer. Exposure control system for an electronic imaging camera having increased dynamic range, June 1987.
- [106] J. Serot N. Roudel, F. Berry and L. Eck. Hardware implementation of a real time lucas and kanade optical flow. In *Conference on Design and Architectures for Signal and Image Processing*, 2009.
- [107] Zhou Wang and A. C. Bovik. A universal image quality index. *Signal Processing Letters, IEEE*, 9(3) :81–84, 2002.
- [108] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment : from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4) :600–612, 2004.
- [109] Rafał Mantiuk, Scott Daly, Karol Myszkowski, and Hans-Peter Seidel. Predicting visible differences in high dynamic range images - model and its calibration. In Bernice E. Rogowitz, Thrasyvoulos N. Pappas, and Scott J. Daly, editors, *Human Vision and Electronic Imaging X, IS&T/SPIE's 17th Annual Symposium on Electronic Imaging (2005)*, volume 5666, pages 204–214, 2005.
- [110] Florent de Dinechin, Jérémie Detrey, Octavian Cret, and Radu Tudoran. When fpgas are better at floating-point than microprocessors. In *Proceedings of the 16th international ACM/SIGDA symposium on Field programmable gate arrays*, FPGA '08, pages 260–260, New York, NY, USA, 2008. ACM.
- [111] Nicolas Brisebarre and Jean-Michel Muller. Correctly rounded multiplication by arbitrary precision constants. *IEEE Transactions on Computers*, 57(2) :165–174, 2008.
- [112] Nachiket Kapre and Andre DeHon. Optimistic parallelization of floating-point accumulation. In *Proceedings of the 18th IEEE Symposium on Computer Arithmetic*, ARITH '07, pages 205–216, Washington, DC, USA, 2007. IEEE Computer Society.
- [113] J. Detrey and F. de Dinechin. Floating-point trigonometric functions for fpgas. In *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on*, pages 29–34, 2007.
- [114] Florent De Dinechin and Jérémie Detrey. Parameterized floating-point logarithm and exponential functions for FPGAs. *Microprocessors and Microsystems*, 31(8) :537–545, February 2006.
- [115] Yong Dou, S. Vassiliadis, G. K. Kuzmanov, and G. N. Gaydadjiev. 64-bit floating-point fpga matrix multiplication. In *ACM/SIGDA Field-Programmable Gate Arrays*, pages 86–95. ACM Press, 2005.
- [116] B. Lee and N. Burgess. Parameterisable floating-point operations on fpga. In *Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on*, volume 2, pages 1064–1068 vol.2, 2002.

- [117] Florent de Dinechin, Vincent Lefèvre, Ecole Normale, Supérieure Lyon, Université Mixte, and Recherche Cnrs inria-ens Lyon. Constant multipliers for fpgas, 2000.
- [118] Jianhua Liu, Michael Chang, and Chung-Kuan Cheng. An iterative division algorithm for fpgas. In *Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays*, FPGA '06, pages 83–89, New York, NY, USA, 2006. ACM.
- [119] N. Takagi. A hardware algorithm for computing reciprocal square root. In *Computer Arithmetic, 2001. Proceedings. 15th IEEE Symposium on*, pages 94–100, 2001.
- [120] Steven W. Smith. *The scientist and engineer's guide to digital signal processing*. California Technical Publishing, San Diego, CA, USA, 1997.
- [121] Jérémie Detrey. *Arithmétiques réelles sur FPGA : virgule fixe, virgule flottante et système logarithmique*. PhD thesis, École Normale Supérieure de Lyon, Lyon, France, January 2007.
- [122] Miguel Granados Boris Ajdin Ulm University Michael Wand Christian Theobalt Hans-Peter Seidel Hendrik P. A. Lensch. Optimal hdr reconstruction with linear digital cameras. Technical report, MPI Informatik, 2010.
- [123] Christophe Schlick. Quantization techniques for visualization of high dynamic range pictures. In Georgios Sakas, Stefan Möller, and Peter Shirley, editors, *Photorealistic Rendering Techniques*, Focus on Computer Graphics, pages 7–20. Springer Berlin Heidelberg, 1995.
- [124] A. Zuloaga, J. L. Martin, U. Bidarte, and J. A. Ezquerro. Vhdl test bench for digital image processing systems using a.
- [125] Daniel Tamburrino, David Alleysson, Laurence Meylan, and Sabine Suesstrunk. Digital camera workflow for high dynamic range images using a model of retinal processing. In DiCarlo, JM and Rodricks, BG, editor, *DIGITAL PHOTOGRAPHY IV*, volume 6817 of *Proceedings of SPIE*, 1000 20TH ST, PO BOX 10, BELLINGHAM, WA 98227-0010 USA, 2008. Soc Imaging Sci & Technol ; SPIE, SPIE-INT SOC OPTICAL ENGINEERING. Conference on Digital Photography IV, San Jose, CA, JAN 28-29, 2008.
- [126] Marcos Slomp and Manuel M. Oliveira. Real-time photographic local tone reproduction using summed-area tables.
- [127] O. Gallo, N. Gelfand, W. Chen, M. Tico, and K. Pulli. Artifact-free high dynamic range imaging. In *Proceedings of the IEEE International Conference of Computational Photography (ICCP)*, San Francisco, 04/2009 2009.
- [128] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment : from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4) :600–612, 2004.
- [129] BERRY F. ABID M., DIAS REAL F. and SEROT J. Harnessing a multi-sensor fpga-based smart camera : a virtual processor-based approach. In *ACTI*, 2009.

# TABLE DES FIGURES

1.1	Analogie entre photographie classique et perception visuelle humaine du point de vue du peintre. Illustration des limites de la photographie numérique au travers d'une image résultante d'une capture classique (a), et d'une peinture de Pierre Bonnard (b). . . . .	13
1.2	Exemple d'une série d'images prises avec des temps d'exposition différents, technique aussi appelée "bracketing". Les images sont tirées du travail de Fattal et al. [17]. . . . .	14
1.3	Image HDR après tone mapping, réalisée à partir de la séquence d'images de la Figure 1.2. . . . .	14
2.1	Exemple d'image HDR issue du livre de Debevec [23]. On remarque que l'on peut distinguer à la fois les hautes luminosités provenant de l'extérieur, et les basses luminosités de l'intérieur de la pièce. . . . .	21
2.2	Applications de l'imagerie HDR. Une opération chirurgicale (a) et un guidage de soudure (b). . . . .	22
2.3	Illustration de la mesure de radiance ou de luminance. Elle représente le flux de puissance électromagnétique/lumineuse par unité de surface, en provenance ou dans une direction donnée. L'illustration (b) est tirée du livre de Reinhard et al. [23] . . . . .	23
2.4	Les différentes techniques HDR. . . . .	32
2.5	Illustration du système de capture de radiance par une caméra. . . . .	33
2.6	(a) Courbe de réponse inverse $g(Z)$ donnant la relation $g(Z) = \ln E \cdot \Delta t$ , pour le mémoriel de Debevec [10]. . . . .	33
2.7	Quatre fonctions de pondération normalisées [10, 12, 23, 40]. . . . .	36
2.8	Image résultante d'un tone mapping global de type logarithmique. . . . .	38
2.9	Illustration des effets indésirables de type halos, pouvant apparaître lors de l'utilisation de TMO locaux de Chiu et al. [47] (a) et Yee et al. [48] (b). Les images sont tirées du livre de Reinhard [23]. . . . .	39
2.10	Résultats de la qualité d'images produits par l'étude de Cadik et al. [58] . . . . .	41

2.11	Un exemple de la technique de fusion d'exposition proposée par Mertens et al. [64]. Un set d'exposition multiples (a), et un résultat HDR-like sans passer par la création, à proprement parlé, d'une image HDR et par le tone mapping (b). Les images sont de Mertens et al. [64] . . . . .	42
2.12	Illustration de la technique de fusion d'exposition pour des scènes dynamiques. (a) montre une séquence d'acquisition, (b) le résultat de fusion avec la technique originale de Mertens [64] et (c) le résultat de la technique de Pece et al. [67] . . . . .	44
2.13	Différence au niveau des pipelines des méthodes de fusion d'exposition et HDR. . . . .	44
2.14	Une pyramide MTB (Median Threshold Bitmap) (b) est utilisée pour l'alignement des images d'entrée (a) proposé par Ward [68]. . . . .	45
2.15	Exemple d'image HDR issue du livre de Bloch [73]. On peut y décèler le problème de ghost énoncé dans la Section 2.4.2. . . . .	47
3.1	Technique d'extension de dynamique initiée par Dekker et al. [74]. L'image (a) est issue d'une capture linéaire (70 dB), tandis que (b) est une image résultante de la méthode de compression proposée (96 dB). . . . .	50
3.2	La technique de variation spatiale d'exposition. Les pixels peuvent être simultanément échantillonnés afin de capturer toute la gamme de luminance de la scène par petits morceaux. . . . .	51
3.3	Scènes de grande gamme dynamique capturées grâce au capteur logarithmique de Yang Ni et al. [77]. . . . .	52
3.4	Le schéma fonctionnel de la méthode proposée par Tico et al. [80]. . . . .	54
3.5	Résultat du système initié par Tico et al. [80]. (a), (b) et (c) sont l'ensemble des expositions simples LDRs, et (d) est le résultat de leur méthode. . . . .	55
3.6	La ré-exposition partielle par Guthier et al. [84]. Les zones surexposées et sous-exposées sont rassemblées en blocs ROI (cadres bleutés) afin d'être re-capturées. La durée normale d'exposition multiple est ainsi réduite. . . . .	57
3.7	Utilisation d'une CCRF ("Comparametric Camera Response Function") par Ali et al. [85]. L'estimation de la radiance est donnée par $\hat{q} = f_{\Delta EV}^{-1}(f_1, f_2)$ , où $\hat{q}$ est la valeur de radiance, $f$ l'estimateur commun où chaque estimation de radiance dépend de plusieurs mesures. . . . .	58
3.8	Le célèbre mémorial par Debevec et al. [89], utilisé comme image HDR de référence dans le domaine du HDR, au travers de nombreux systèmes logiciels existants. Il est ici tone mappé pour pouvoir l'afficher sur écran standard. . . . .	60

3.9	Le système de casque de soudage "MannVis welding helmet" par Mann et al. [99]. . . . .	66
3.10	Les deux cartes capteur fabriquées au cours de la thèse. Elles supportent les versions de capteurs Sapphire - EV76C560 (1.3 MPixels) et Sapphire - EV76C570 (2 MPixels) de chez e2v. Les cartes sont déclinées en deux versions, une version adaptable pour les connecteurs XGI (a), et une version pour les connecteurs FMC (b). . . . .	68
3.11	HDR-ARtiSt est prototypé sur des carte d'évaluation embarquant un FPGA de type Virtex de chez Xilinx. Les cartes capteur ont été fabriquées par nos soins. Elles accueillent un capteur CMOS EV76C560 1.3 Mégapixels de la société e2v. Les deux différentes "cartes mères" ici sont la ml507 (a) et la ml605 (b). . . . .	73
4.1	Vue globale du système HDR-ARtiSt. . . . .	77
4.2	La Figure (a) est composée 15,3% de pixels saturés blanc lorsque la lampe est allumée. Après l'opération d'auto exposition, la Figure (b) présente 5,4% de pixels saturés blancs. . . . .	80
4.3	La Figure (a) est composé 38,3% de pixels saturés noir lorsque la lampe est allumée. Après l'opération d'auto exposition multiple, la Figure (b) présente 0,2% de pixels saturés dans le noir. . . . .	81
4.5	Comparaison entre les framerates de la technique HDR standard et de notre technique. Un pipeline d'images est mis en place afin de délivrer à chaque instant plusieurs flux vidéos en parallèle. . . . .	85
4.6	Initialisation de la MMU, avec le stockage des $P-1$ images dans la mémoire et la lecture des $P-1$ premières lignes des $P-1$ premières images. . . . .	86
4.7	Initialisation du management de la mémoire. Le capteur envoie séquentiellement une exposition basse ( $I_L$ ) et moyenne ( $I_M$ ). On peut noter l'écriture dans la mémoire des $M$ lignes, pour les deux premières images. . . . .	87
4.8	Unité de gestion de mémoire avec l'acquisition nouvelles lignes de pixels et la lecture en parallèle des lignes précédemment stockées. . . . .	88
4.9	HDR-MMU. Gestion de trois flux vidéos parallèles des images $I_1$ , $I_2$ et $I_3$ d'exposition basse, moyenne et haute dans le cas où $P = 3$ . La sortie HDR est sujette à une latence algorithmique (partie grisée). . . . .	89
4.10	Schéma global de l'organisation de test des algorithmes HDRs. . . . .	90
4.11	Utilisation des ressources pour les opérateurs d'addition et de multiplication sur Virtex 6. . . . .	96
4.12	Illustration tirée du livre de Steven et al. [120]. . . . .	97

4.13	Tableau de bord du LogiCORE de chez Xilinx. Implantation d'un composant arithmétique en flottant IEEE754. Sélection de l'opérateur et optimisations. . . . .	99
4.14	Courbe de réponse inverse $g(Z_{ij}) = \ln(E_i \Delta t_j)$ de notre système HDR-ARtiSt, créée à partir de 5 images LDRs et de l'algorithme de Debevec. Voir l'Annexe C pour plus de détails. . . . .	101
4.15	Les quatre fonctions de pondération en test. l'algorithme de visualisation tone mapping est celui de Duan et al. [50]. . . . .	101
4.16	Création du pipeline HDR pour la vidéo HDR-3, en utilisant une architecture en cascade. Des LUTs permettent de stocker la courbe de réponse ainsi que des valeurs pré-calculées de gamme de logarithmes. . . . .	103
4.17	Les trois images LDRs (a), (b) et (c) qui nous ont servi à effectuer des comparaisons entre les TMOs. (d) et (e) sont respectivement l'image de contribution maximale et l'image fausse couleur des luminances de l'image HDR. . . . .	105
4.18	Résultat de comparaison des algorithmes de tone mapping [13, 14, 50] sous HDR-Lab. . . . .	106
4.19	Résultats de HDR-VDP 2 [109] ("Visible Difference Metric for High Dynamic Range Images") sur les algorithmes sélectionnés dans HDR-Lab. (a) : Tumblin [45], (b) : Drago [13], (c) : Schlick [123], (d) : Reinhard local [14], (e) : Reinhard global [14], (f) "Exposure Fusion" [64], (g) : Duan [50]. . . . .	107
4.20	Image HDR après tone mapping de Duan et al. [50] à partir de trois images LDRs. La carte de radiance a été construite avec l'algorithme de Debevec et al. [10]. Les trois images LDRs sont tirées d'Internet. . . . .	108
4.21	Calcul de tone mapping par le méthode de Duan et al. [50]. . . . .	109
4.22	Calcul de tone mapping par le méthode de Reinhard et al. [14]. . . . .	109
5.1	Schéma du bloc de testbench hardware. . . . .	113
5.2	Évaluation visuelle de l'application HDR-2 ( $P = 2$ ) en simulation hardware avec le mémorial de Paul Debevec. L'image fausse couleur (d) des valeurs de radiances réelles capturées avec 16 images LDRs dans la revue de Debevec [10], montre les zones de la Figure (c) où la quantité d'information est limitée. . . . .	115
5.3	Comparaison entre le système HDR-2 et HDR-3. On y aperçoit les améliorations apportées par le système vidéo HDR-3 dans des conditions d'éclairage extrêmes. . . . .	120

5.4	Comparaison entre les versions software et hardware du pipeline HDR-3 (Debevec+Duan). . . . .	121
5.5	Images N & B (a) et couleurs (b) et (c) issues du capteur couleur de chez E2v, et les résultats HDRs obtenus avec le tone mapping de Duan et al. (d) et Reinhard et al. (e). . . . .	123
5.6	Aperçu de l'application HDR-ViewFinder réalisée sous Qt, permettant l'interaction avec la caméra HDR-ARtiSt via le protocole UDP par liaison Ethernet. . . . .	124
5.7	Comparaison de la sortie du système proposé avec les différentes méthodes de tone mapping en temps-réel existantes. . . . .	126
5.8	Ghost apparaissant au sein du système HDR-ARtiSt. . . . .	129
A.1	Schéma global de l'interface mémoire du système HDR-ARtiSt. Elle est composée de la MIG de Xilinx, de notre MMU et du pipeline de BRAMs. . .	135
A.2	Schéma du pipeline de BRAMs de la MMU pour $P = 3$ . Le nombre de BRAMs nécessaires au système général à $P$ images est de $2P$ . . . . .	137
A.3	Schéma simplifié de la machine à états de la MMU pour $P = 3$ . . . . .	138
B.1	Dissection des parties du logiciel <i>HDR-Lab</i> sous Matlab. . . . .	140
C.1	Images LDRs issues de notre camera <i>HDR-ARtiSt</i> , utilisées afin d'estimer la courbe de réponse du système à partir de logiciel <i>HDR-Lab</i> . Les temps d'exposition sont : $16\mu s$ , $110\mu s$ , $629\mu s$ , $2.53ms$ et $9.00ms$ . Les images de contribution et fausse couleur associées sont représentées (f) et (g). . . . .	143
D.1	Les différentes sorties parallélisées du système HDR-ARtiSt. . . . .	146
E.1	L'étalon or pour les comparaisons d'algorithmes de tone mapping sous HDR-Lab. C'est une version d'image HDR créée à partir de 16 images LDRs, et tone mappée avec l'algorithme de Drago et al. [13]. . . . .	148
F.1	Schéma de la carte fille destinée à être montée sur des connecteurs XGI. .	151
F.2	Schéma de la carte sous-fille destinée à être montée sur la carte fille. . .	152
F.3	Vues recto verso des cartes capteur fille et sous-fille. . . . .	153





# LISTE DES TABLEAUX

1.1	Collaboration avec les entreprises et les services au cours de la thèse. . . . .	15
2.1	Les grandeurs radiométriques et photométriques. . . . .	25
2.2	Liste des notations fréquemment utilisées dans cette thèse. . . . .	26
2.3	Valeurs et correspondances de dynamiques en gammes de contrastes, decibels et stops. . . . .	27
2.4	Comparaison CCD VS CMOS. Images tirées de white paper de chez Grass Valley . . . . .	29
2.5	Liste des fonctions de pondération existantes. . . . .	36
2.6	Quelques opérateurs de reproduction de tons. . . . .	40
3.1	Résumé de comparaisons résultant des implémentations de la méthode d'Ureña et al. [96] sur deux architectures différentes. . . . .	64
3.2	Tableau récapitulatif des systèmes matériels vidéo HDR ou HDR-like (ou Exposure Fusion) et des fonctionnalités qu'ils proposent et si ils sont ou non temps-réel (TR). La fonctionnalité de capture suppose la proposition d'une méthode originale permettant de sélectionner judicieusement les temps d'exposition. . . . .	67
3.3	Résumé des particularités des deux FPGAs utilisés au cours de la thèse. . . . .	72
4.1	Représentation du nombre flottant 0.99999994 en format IEEE754 simple précision . . . . .	94
4.2	Comparaison de l'estimation des ressources des opérateurs arithmétiques sur Virtex 6, pour une représentation des nombres en virgule fixe et flottant. . . . .	98
4.3	Comparaisons des métriques issues du logiciel HDR-Lab, pour les algorithmes TMOs appliqués à une image HDR commune construite à partir de 3 images. Les temps d'exposition utilisés sont 32s, 1s et 31,25ms. . . . .	104
5.1	Résumé des caractéristiques communes aux deux prototypages sur ml507 et ml605 (Virtex 5/6). . . . .	118

5.2	HDR-2 : Résumé du rapport d'implémentation matériel sur Virtex 5 . . . . .	118
5.3	HDR-3 : Résumé du rapport de synthèse hardware sur Virtex 5. . . . .	122
5.4	Résumé de l'implémentation HDR-2 et HDR-3 pour la plate-forme Virtex 6. . . . .	124
5.5	Comparaisons avec les architectures existantes de tone mapping en temps réel. . . . .	127



## Résumé :

Cette thèse s'intègre dans le cadre du projet européen EUREKA "High Dynamic Range - Low Noise CMOS imagers", qui a pour but de développer de nouvelles approches de fabrication de capteurs d'images CMOS à haute performance. L'objectif de la thèse est la conception d'un système de vision temps réel à grande gamme dynamique (HDR). L'axe principal sera la reconstruction, en temps réel et à la cadence du capteur (60 images/sec), d'une vidéo à grande dynamique sur une architecture de calcul embarquée.

La plupart des capteurs actuels produisent une image numérique qui n'est pas capable de reproduire les vraies échelles d'intensités lumineuses du monde réel. De la même manière, les écrans, imprimantes et afficheurs courants ne permettent pas la restitution effective d'une gamme tonale étendue. L'approche envisagée dans cette thèse est la capture multiple d'images acquises avec des temps d'exposition différents permettant de palier les limites des dispositifs actuels.

Afin de concevoir un système capable de s'adapter temporellement aux conditions lumineuses, l'étude d'algorithmes dédiés à la grande dynamique, tels que les techniques d'auto exposition, de reproduction de tons, en passant par la génération de cartes de radiances est réalisée. Le nouveau système matériel de type "smart caméra" est capable de capturer, générer et restituer du contenu à grande dynamique dans un contexte de parallélisation et de traitement des flux vidéos en temps réel.

## Abstract:

This thesis is a part of the EUREKA European project called "High Dynamic Range - Low Noise CMOS imagers", which developed new approaches to design high performance CMOS sensors. The purpose of this thesis is to design a real-time high dynamic range (HDR) vision system. The main focus will be the real-time video reconstruction at 60 frames/sec in an embedded architecture. Most of the sensors produce a digital image that is not able to reproduce the real world light intensities. Similarly, monitors, printers and current displays do not recover of a wide tonal range. The approach proposed in this thesis is multiple acquisitions, taken with different exposure times, to overcome the limitations of the standard devices.

To temporally adapt the light conditions, the study of algorithms dedicated to the high dynamic range techniques is performed. Our new smart camera system is able to capture, generate and show content in a highly parallelizable context for a real time processing.

